26898

National Library
of Canada

Bibliothèque nationale
du Canada

CANADIAN THESES
ON MICROFICHE

THÈSES CANADIENNES
SUR MICROFICHE

NAME OF AUTHOR/*NOM DE L'AUTEUR* JOHN RAMSDEN

TITLE OF THESIS/*TITRE DE LA THÈSE* NUMERICAL METHODS IN FABRIC ANALYSIS

UNIVERSITY/*UNIVERSITÉ* UNIVERSITY OF ALBERTA

DEGREE FOR WHICH THESIS WAS PRESENTED/
GRADE POUR LEQUEL CETTE THÈSE FUT PRÉSENTÉE Ph.D.

YEAR THIS DEGREE CONFERRED/*ANNÉE D'OBTENTION DE CE GRADE* 1975

NAME OF SUPERVISOR/*NOM DU DIRECTEUR DE THÈSE* D. M. CRUDEN

Permission is hereby granted to the NATIONAL LIBRARY OF CANADA to microfilm this thesis and to lend or sell copies of the film.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

L'autorisation est, par la présente, accordée à la BIBLIOTHÈQUE NATIONALE DU CANADA de microfilmer cette thèse et de prêter ou de vendre des exemplaires du film.

L'auteur se réserve les autres droits de publication; ni la thèse ni de longs extraits de celle-ci ne doivent être imprimés ou autrement reproduits sans l'autorisation écrite de l'auteur.

DATED/*DATÉ* 10 October 75 SIGNED/*SIGNÉ* John Ramsden

PERMANENT ADDRESS/*RÉSIDENCE FIXE* #202 10710-85 Av.
Edmonton, Alberta
Canada

NL-91 (3-74)

THE UNIVERSITY OF ALBERTA

NUMERICAL METHODS IN FABRIC ANALYSIS

by

JOHN RAMSDEN, B.A.Sc., M.Sc. ©

A THESIS
SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH
IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE
OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF GEOLOGY

EDMONTON, ALBERTA
FALL, 1975

UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled "Numerical Methods in Fabric Analysis", submitted by John Ramsden, B.A.Sc, M.Sc., in partial fulfilment of the requirements for the degree of Doctor of Philosophy.

....
Supervisor

H.A.K. Charlesworth
....

....

....

F.P. Agterberg
....
External Examiner

Date 6 Oct 1975

Abstract

This thesis describes an approach to fabric analysis that differs from the conventional approach in making more extensive use of numerical methods in all stages \of the procedure. Some new numerical and display techniques are presented; some techniques are described that are not in themselves new but have not been used before in fabric analysis; a procedure for computer-assisted fabric analysis has been devised, and all the computer programs necessary to implement the procedure are provided.

The widespread adoption of numerical methods of producing contoured orientation diagrams in place of the conventional graphical methods leads to consideration of estimators of population density other than the one conventionally used by geologists because of its unique adaptability to graphical solution. The existing theory of nonparametric density estimators shows the conventional estimator to be but a special case of one class of density estimators. Exploratory investigation (theoretical and experimental) of a different estimator of the same class and of an estimator of a different class under simple conditions of unimodal Fisher-distributed populations reveals differences in the statistical properties of the estimators, the greatest differences appearing between estimators of different classes.

Contoured density diagrams are used to suggest a

tentative model for the population and to provide estimates of model parameters. For both purposes a series of diagrams should be produced using a wide range of degrees of smoothing. No particular degree of smoothing can be chosen _a priori_ as being optimum for either purpose, since the optimum degree of smoothing in both cases depends on the population and may, furthermore, be different for different elements of the population. When the general class of models being considered is a mixture of Fisher distributions, estimates of concentration parameters can be obtained from measures of the size and/or shape of the peaks appearing on contoured density diagrams using relationships derived theoretically and verified experimentally in this study. The contoured density diagram thus serves in the model-selection and parameter-estimation stages of the model-building process; except in the case of simple single-element populations, techniques for the remaining stage - testing the adequacy of the model - are lacking, and a model thus constructed cannot be regarded as statistically sound.

Techniques are described for characterizing an isolated fabric element. Such a characterization is meaningless unless its adequacy is tested using a combination of data displays and statistical tests; suitable tests are described. Multisample tests are described for axially symmetric cases. Most of the characterization techniques and tests for axis data are new to fabric analysis. A new

technique for producing smoothed rose diagrams is described, and ways of dealing with weighted observations suggested.

A procedure is described for definition of homogeneous subfabric domains based on statistical tests for homogeneity of mean directions and dispersions of local samples. Local disturbances of the fabric are identified by exceptionally large dispersions or exceptionally large deviations from the overall mean. A new map display simultaneously shows location and three-dimensional orientation of fabric elements or local means, or deviations of these from an overall mean, so that regions of consistent deviation from the overall mean can be readily identified as distinct fabric domains.

Table of Contents

1 Introduction 001
  1.1 Conventional Fabric Analysis 001
    1.1.1 Definitions 001
    1.1.2 Reasons for Studying Fabric 003
    1.1.3 Conventional Fabric Analysis 004
  1.2 Computer-Assisted Fabric Analysis 005
    1.2.1 Drawbacks of the Conventional Procedure 005
    1.2.2 Advantages of Computer-Assisted Procedure 005
    1.2.3 New Procedure Using the Computer 007
    1.2.4 Designing a Computer-Assisted Procedure 008
    1.2.5 Scope and Organization of Thesis 010
  1.3 Data Collection, Storage and Retrieval 011
    1.3.1 Data Collection 011
    1.3.2 Data Organization and Storage 013
    1.3.3 Data Retrieval 014

D

2 Data Display - Contoured Orientation Diagrams 019
  2.1 Introduction 019
  2.2 Review of Contouring Methods 020
  2.3 Two New Contouring Methods 023
    2.3.1 Nature of the Contouring Process 023
    2.3.2 The Conventional Constant-Area Method 024
    2.3.3 The Fisher-Weighted Method 027
    2.3.4 The Constant-Proportion Method 029
  2.4 Theoretical Investigation of Contouring Methods ...030
    2.4.1 The Constant-Area Method 030
    2.4.2 The Fisher-Weighted Method 034

2.4.3 The Constant-Proportion Method 035
2.5 Experimental Investigation of Contouring Methods ..036
2.5.1 The Experiments 036
2.5.2 Results and Discussion 038
2.6 Use of the Orientation Diagram 044
2.6.1 The Modelling Process 044
2.6.2 Estimation of Fisher Concentration Parameters 045
2.6.3 Optimization of Density Estimators 048
2.7 Summary and Conclusions 052
3 Numerical Characterization 057
3.1 Introduction 057
3.2 Descriptive Statistics 061
3.2.1 Distributions on the Sphere 061
3.2.2 Vector Statistics 062
3.2.3 Axis Statistics 062
3.3 Distribution Models 063
3.3.1 Fisher's Distribution 063
3.3.2 Bingham's Distribution 065
3.3.3 Dimroth-Watson's Distribution 066
3.4 Testing the Adequacy of the Model 068
3.4.1 General Statement 068
3.4.2 Vector Statistics 070
3.4.3 Axis Statistics 072
3.5 Treating Weighted Observations 075
3.6 Computer Programs 077
3.7 Numerical Examples 082
3.8 Summary and Conclusions 085

4 Areal Analysis 088
  4.1 Introduction 088
  4.2 Multisample Tests 092
    4.2.1 Fisher's Model 092
    4.2.2 Dimroth-Watson's Model 095
  4.3 Computer-Based Procedure 097
  4.4 Worked Example 100
  4.5 Summary and Conclusions 102
5 Summary 104
  5.1 General Statement 104
  5.2 Data Collection, Storage and Retrieval 104
  5.3 Data Display - Contoured Orientation Diagrams 105
  5.4 Numerical Characterization 107
  5.5 Areal Analysis 108
Tables and Figures 110
References Cited 155
Glossary of Symbols 163
Appendix 1: Documentation of Computer Programs 167
Appendix 2: Documentation of Subroutine Library 381
Appendix 3: Table of Density Estimator Data 422

8

List of Tables

Table Page

1 Output from Program MODEL1 for Fording Domain 3 Bedding 110

2 Sample Sizes and Eigenvalues for the Till Samples ..111

3 Kolmogorov Test-of-Fit of 10 Values of the Statistic [4.3] to the F-Distribution with 38 and 360 Degrees of Freedom 112

4 Kolmogorov Test-of-Fit of 10 Values of the Statistic [4.2] to the Chi-Square Distribution with 38 Degrees of Freedom 113

5 Kolmogorov Test-of-Fit of 10 Values of the Statistic [4.2] to the Chi-Square Distribution with 19 Degrees of Freedom 114

6 Sizes, Concentration Parameters, and Traverse Numbers for Initial Domains, Fording Domain 3 Bedding 115

List of Figures

Figure Page

1 Procedure for Computer-Assisted Fabric Analysis 116

2 Program Inputs and Outputs for Data Collection, Storage, Retrieval, Display and Characterization Operations 117

3 Simplified Flow Diagram for Program KEY4 118

4 Properties of a Binomial Variable in Terms of p and n 119

5 Fisher's Density Function for k=50 120

6 Parameter Curves for the Constant-Area Estimator 121

7 Parameter Curves for the Constant-Proportion Estimator 122

8 Parameter Curves for the Fisher-Weighted Estimator 123

9 Location Curves for the Constant-Area Estimator 124

10 Location Curves for the Constant-Proportion Estimator 125

11 Location Curves for the Fisher-Weighted Estimator 126

12 Location Curves for Gamma = 25 degrees 127

13 Location Curves for Gamma = 20 degrees 128

14 Location Curves for Gamma = 15 degrees 129

15 Location Curves for Gamma = 10 degrees 130

16 Location Curves for Gamma = 5 degrees 131

17 Location Curves for Gamma = 0 degrees 132

18 Effect of Sample Size on the Constant-Area Estimator 133

19 Effect of Sample Size on the Constant-Proportion Estimator 134

20 Effect of Sample Size on the Fisher-Weighted Estimator 135

21 Effect of Population Parameter on the Constant-Area Estimator for Gamma = 0 degrees 136

22 Effect of Population Parameter on the Constant-Proportion Fisher-Weighted Estimators for Gamma = 0 degrees 137

23 The Model-Building Process 138

24 Spherical Point Distributions 139

25 Lower-Hemisphere Equal-Area Projection of 297 Poles to Bedding in Domain 3 at the Fording Coal Mine 140

26 Displays for Sample 30; 90 observations 141

27 Displays for Sample 100; 49 observations 142

28 Displays for Sample 110; 50 observations 143

29 Procedure for Areal Analysis - Part 1 144

30 Procedure for Areal Analysis - Part 2 145

31 Procedure for Areal Analysis - Part 3 146

32 Procedure for Areal Analysis - Part 4 147

33 Procedure for Areal Analysis - Part 5 148

34 Procedure for Areal Analysis - Part 6 149

35 Procedure for Areal Analysis - Part 7 150

36 Lower-Hemisphere Equal-Area Projection of 297 Poles to Bedding in Domain 3 at the Fording Coal Mine after Rotation to Make the Mean Vertical 151

37 Fording Bedding Data - Locations and Orientations of Observed Discontinuities 152

38 Fording Bedding Data - Traverses with Numbers of Observations (above) and Traverse Numbers (below) .. 153

39 Fording Bedding Data - Deviations of Domain Means from Overall Mean 154

1 Introduction

1.1 Conventional Fabric Analysis

1.1.1 Definitions

The internal geometric properties of a rock body are collectively referred to as the fabric of the body. In studying fabric an investigator observes a large number of individual structures, which may be planar structures such as joints, bedding and cleavage, or linear structures such as long axes of sedimentary particles or intersections of bedding and cleavage. In order to form part of the fabric, a structure must be penetrative, that is, it must be present throughout the body of rock being studied. Thus a single shear surface is not part of the fabric of the body, but many closely spaced shear surfaces throughout the body are. Each kind of penetrative structure with a distinct geometric configuration is called a fabric element; the geometric configuration of a particular fabric element is called the subfabric for that element. The fabric is the sum of all the subfabrics (Turner and Weiss, 1963, pp.15-45).

The primary object of fabric analysis, and the one with which we are exclusively concerned in this thesis, is to describe the fabric of the rock body under study. For this purpose the rock body is subdivided into the smallest number of portions within which the fabric may be regarded as spatially homogeneous. Each such portion is called a fabric domain, and the fabric of each domain is described separately.

What constitutes a fabric domain depends on the scale of the investigation. "Scale" here does not refer to the geographic extent of the rock body under study, but rather to the size of the smallest portion of the rock body that is considered as a distinct unit, referred to here as a "sampling unit". For example, in a very small-scale study, the sampling unit might be a thin section of rock examined microscopically. In this case, a portion of the rock body would be a fabric domain only if all the thin sections from within that portion had similar fabrics. On the other hand, an entire outcrop might be treated as a sampling unit if variations on a smaller scale were not of interest. In this case a fabric domain would be a group of adjacent outcrops having similar fabrics. Note that if the locations of all observations are precisely recorded, the sampling units can later be redefined, either on the same scale or on a larger or smaller scale (redefinition of sampling units on a smaller scale being limited by the density of observations). Finally, we define a subfabric domain for a particular fabric element as the largest group of adjacent sampling units having similar subfabrics for that fabric element.

1.1.2 Reasons for Studying Fabric

Suppose a geologist wishes to elucidate the history of deformation of a body of deformed rock, to obtain information about the forces that caused the deformation. He makes the reasonable assumption that the observed fabric of the rock reflects the stresses and strains that have been a part of the history of the rock. Or a geologist may wish to determine the direction of movement of the transporting medium that deposited a sediment; for example, the direction of wind that formed a sand dune (the prevailing wind direction is important in inferring the latitude and/or orientation of the land mass at the time of deposition of the sediment), the direction of flow of a river that deposited a sand bar (if the bar contains oil, other pools may be found in the upstream or downstream directions), the direction of movement of a glacier that deposited a till (regional flow patterns of ancient continental glaciers have important implications for the study of past and present climates). Again the assumption is made (well substantiated in both field and laboratory) that the present fabric of the rock or sediment is inherited from the flow of the transporting medium, with more or less modification by post-depositional processes. An engineer may wish to determine safe angles of rock slopes to reduce the risk of slope failure and to reduce the cost of removing rock and overburden in making the slope. The underlying assumption here is that the type of failure and probability of failure of a rock slope depend on the orientation, density and physical properties of the discontinuities within the rock.

1.1.3 Conventional Fabric Analysis

The conventional procedure for fabric analysis is described by Turner and Weiss (1963, pp.146-151), and consists of the following steps.

(1) Collect data as uniformly as possible over the exposed parts of the rock body.

(2) Subdivide the rock body into sampling units.

(3) Plot orientation diagrams for all measured subfabrics for all sampling units.

(4) By trial and error, combine adjacent sampling units having similar fabrics (that is, sampling units all of whose subfabrics are similar) into fabric domains.

(5) Look among the fabric domains for subfabric domains which are even more extensive.

1.2 Computer-Assisted Fabric Analysis

1.2.1 Drawbacks of the Conventional Procedure

The conventional method of fabric analysis suffers from the following disadvantages.

(1) All the fabric elements measured must be identified and distinguished at the beginning of the study, and each structure must be identified at the time it is measured as belonging to one particular fabric element, so that each fabric element can be plotted separately for each sampling unit.

(2) In order to study the fabric on a scale different from that initially chosen, a great deal of time and work must be invested in redefining the sampling units and replotting the diagrams.

(3) Combination of sampling units by trial and error is tedious and time-consuming. Moreover it is a qualitative and subjective procedure, lacking in statistical criteria.

1.2.2 Advantages of Computer-Assisted Procedure

Use of the computer in fabric analysis provides the following advantages.

(1) Orientation diagrams can be plotted rapidly and cheaply by the computer even for large numbers of data

points. Thus it is feasible, using the computer, to make diagrams showing all the observations of a particular fabric element, or all the observations of all the fabric elements. This aids in determining the number of fabric elements present and removes the necessity of identifying each structure at the outcrop, since with such diagrams available the different fabric elements can be more reliably separated after all the observations have been made.

(2) If the data are stored in machine-readable form the computer can selectively retrieve the data, so that orientation diagrams for each fabric element in each sampling unit can be produced easily and quickly.

(3) When the data are handled by the computer, it is easy to implement statistical tests for similarity of subfabrics in different sampling units, thus removing much tedious work and providing a quantitative basis for decisions.

(4) The entire analysis is speeded up to such a degree that many iterations are possible, allowing more thorough testing of more hypotheses, thus increasing the value of the analysis.

(5) Due to the high speed of the computer, techniques can be used freely that were either laborious or not feasible at all in conventional fabric analysis. For example, numerical methods of producing contoured

orientation diagrams, that have many advantages over graphical methods, are already widely implemented using computers; without a computer only graphical methods are feasible. As another example, rotation and replotting of fabric data can often aid in the analysis. While simple to achieve using the computer, the conventional graphical rotation procedure is so laborious as to be used only when unavoidable, especially when many observations are involved.

1.2.3 New Procedure Using the Computer

As indicated above, a somewhat different procedure for fabric analysis can be followed when a computer is used. This new procedure is illustrated in Figure 1. The first step in this procedure after collection, organization and storage of the data is selective retrieval of those data thought to constitute one fabric element, for example, observations of joints. The retrieved data is characterized and displayed. The numerical characterization aids in the next step, which is to decide whether or not the retrieved data actually constitute a single fabric element. If more than one fabric element is present, the definition is made more restrictive, for example, joints with calcite fillings, and retrieval, characterization and display are repeated. After a fabric element has been isolated, it is necessary to perform an areal analysis to define the subfabric domains.

Areal analysis begins with the production of map displays by the computer, to be used by the geologist for outlining the sampling units. Once the information defining these has been made available to the computer, the subfabric of each sampling unit is characterized and more map displays are produced showing the subfabrics. The geologist uses these displays to group similar samples, and the computer performs statistical tests for homogeneity of the grouped samples. An interactive program allows this operation to be performed many times as rapidly as the geologist can enter new groupings. Thus the domains for that subfabric can be quickly defined. As each domain is defined, its subfabric is characterized.

The next fabric element is then retrieved from the data file and the process repeated. After all the fabric elements have been identified, and subfabric domains defined for each, fabric domains are found by superimposing the subfabric domains. Thus, in this procedure, subfabric domains are established first, and then fabric domains obtained by combining these.

1.2.4 Designing a Computer-Assisted Procedure

The steps involved in designing a scheme for computer-assisted fabric analysis were as follows.

(1) The entire fabric analysis procedure was broken down into the sequence of operations and decisions shown in Figure 1. This established the operations involved, the

sequence in which these operations would be performed, and the loops or re-entry points required for iterative operations or sequences of operations.

(2) The operations shown in Figure 1 were categorized according to whether or not they could be performed by the computer. Operations involving only data handling, computation or production of displays can be performed by the computer; those involving decisions based on experience and judgement cannot. The operations represented by rectangular boxes on Figure 1 can be performed by the computer; those represented by trapezoidal boxes must be performed by the geologist. The decisions controlling the sequence of these operations are of course made by the geologist.

(3) The arrangement of computer programs and the inputs and outputs of each was then designed, taking into account flexibility, efficiency and possibility of re-entry at various points as well as the essential inputs and outputs. The final arrangement of the programs and their inputs and outputs is shown in Figure 2 for the data organization, storage, retrieval, characterization and display operations. Programs, inputs and outputs for the remaining operations, comprising the areal analysis phase of the procedure, are detailed in Chapter 4.

(4) The computer programs were written. These are documented in Appendix 1.

1.2.5 Scope and Organization of Thesis

The contributions of this thesis to the methodology of fabric analysis are of four kinds.

(1) Some new numerical and display techniques are presented. These are (a) two new methods of contouring point density, including experimental and theoretical investigation of the new and conventional methods, and (b) a new map display to assist in areal analysis.

(2) Some techniques are presented that are not in themselves new but have not been used before in fabric analysis. These are statistical tests for axial orientation data. Some procedures for weighted observations are suggested.

(3) A scheme for computer-assisted fabric analysis has been devised. In this scheme all operations involving decisions are carried out by the geologist, while the computer performs all data handling, computation and display operations.

(4) All the computer programs necessary to implement the above computer-assisted procedure are provided.

The organization of the thesis follows the outline of Figure 1. The remainder of this chapter deals with data collection, organization, storage and retrieval operations. Chapter 2 is concerned with contoured orientation diagrams, and includes a description of two new methods of estimating populations. Chapter 3 reviews available statistical methods of characterizing isolated fabric elements, and introduces some techniques for dealing with weighted observations. The remaining operations shown in Figure 1, that is, the production of map displays, calculation of statistics, and tests for homogeneity of subfabrics are discussed in Chapter 4. Chapter 5 contains a brief summary and a list of recommendations.

1.3 Data Collection, Storage and Retrieval

1.3.1 Data Collection

Two sets of data were used in the development of the methods described in this thesis. One consists of observations of discontinuities of various kinds made in several open-pit mines in western Canada. Sampling was by means of continuous traversing. Straight-line traverses were laid out along the pit benches, and all discontinuities intersecting each traverse were measured. The location and orientation of each traverse, and the distance of each discontinuity along the traverse, were recorded. As well as the orientation of each discontinuity, various physical properties were recorded, such as lithology, size, fillings, water and waviness. Linear fabric elements were always recorded as a pitch on a discontinuity surface.

The other set of data consists of the orientations of the long-axes of stones in glacial till. The stones had sizes (long-axis lengths) from about 1cm to about 10cm and ratios of long-axis length to intermediate-axis length from about 1.5 to about 2.0. Dimensions of the stones were not recorded. Samples of 50 to 100 axes were measured at selected sites within the exposure, the dimensions of the sample sites not exceeding about 0.5m in the direction normal to the bounding surfaces of the till layer and about 2m in the direction parallel to the bounding surfaces of the till. Locations of the sample sites were recorded.

Input of the data to the computer is greatly facilitated if the data are recorded in the field on special field data sheets that conform to the requirements of the study. This has the additional advantages that, first, data collection is standardized in case several field personnel are involved, and, second, in the course of drawing up the field data sheets the investigator is forced to plan carefully the requirements of the survey in terms of the various aspects of data collection: what fabric elements are of interest, what rock types are likely to be encountered, what associated physical properties should be observed and recorded, what sampling schemes are appropriate (or possible), and so on.

Many such field data sheets have been devised for various purposes. Some of these are described by Haugh, Brisbin and Turek (1967) (for structural data), Berner et al (1972) (for general geological mapping), Laurin et al (1972) (for geological mapping in the Precambrian shield), Roddick and Hutchison (1972) (primarily for lithologic data), and Lambert and Reesor (1974) (for all kinds of geological data including structural data). The method of collecting and storing the data should not limit the flexibility of retrievals.

1.3.2 Data Organization and Storage

Before storage, the fabric data were reduced to a form suitable for subsequent computer processing. Since calculations involving the data are most conveniently performed using direction cosines, the orientations of linear fabric elements were reduced to the direction cosines of the line, while the orientations of planar fabric elements were reduced to the direction cosines of their normals (poles). For the open pit data, a correction factor was computed for each observed discontinuity to compensate for the sampling bias associated with straight-line traversing, as follows. The number of planes of a given set intersected by unit length of traverse (the apparent density of the planes) depends not only on the true density of the planes but also on the angle the planes make with the traverse. A simple geometric argument shows that the apparent density of parallel planes in a direction making an angle β with the planes is less than the true density by a factor $\sin\beta$ (Terzaghi, 1965). Thus, to obtain estimates of the relative true densities of planes in various orientations, the observed densities are corrected by the factor $1/\sin\beta$. This is achieved in practice by associating each observed plane with a correction factor or weight of $1/\sin\beta$. Extremely large weights are avoided by truncating at $\beta = 5$ degrees, or $1/\sin\beta \approx 11.5$. Thus the range of weights is from 1.0 to about 11.5.

A simple file structure was adopted for the fabric data, in which all records of the data file had the same format. For the discontinuity data, each record contained the information relating to one observation of a discontinuity. Data organization and storage for this data was performed by the program CONVERT3, documented in Appendix 1. A similar file structure was adopted for the pebble axis data; each record of this file contained the information relating to one axis.

1.3.3 Data Retrieval

It was found after some experience that retrieved subsets of the data file were usually subjected to further analysis, and that it was usually not possible to predict at the time of the retrieval which data items would be required in the subsequent analysis. For this reason the retrieval program was designed to retrieve entire data records rather than individual data items. This had the advantage of simplifying the retrieval program.

To allow for changes in the format of the data file, and to permit the retrieval program to be used with different data files with different record formats, the retrieval program was designed to be independent of the record format of the data file. This was achieved as follows. Each field of the data record containing one data item is given a unique name and a table is constructed containing, on each line of the table, one data name and the FORTRAN format specification for reading the corresponding field. This table is made available to the retrieval program, and data items are referred to by name in the retrieval instructions. From the table the program obtains the format specification.

Substantial simplification of the retrieval program was achieved by limiting the program to read only one field of the data record on each "pass" over the data file. This means that a retrieval requires as many passes over the data file as there are data items involved in the retrieval request. Since most retrievals involve conditions on only one or two data items, and since much less computer time is required to read one field than to read the entire record, this limitation not only simplified the retrieval program but alsc made retrievals faster.

Thus the retrieval program (KEY4, Appendix 1) accepts two kinds of instructions (Figure 3). One kind of instruction involves a condition on a named data item and causes the program to read the input data file, storing the record numbers of those records satisfying the condition. This set of record numbers, called a "key", is associated with a name (given in the instruction) and stored in a temporary file. The other kind of instruction refers (by means of the given names) to previously generated keys, and specifies a combination of the conditions used to produce those keys. The result of this operation is a new key, which is stored in the temporary file.

For example, suppose a file contains structural data and it is required to retrieve all the joints with dip directions less than 45 degrees or greater than 225 degrees. The sequence of instructions, before coding, would be

(1) Generate a key containing the numbers of all the records in which dip direction is less than 45 degrees; call the key "one".

(2) Generate a key for condition "dip direction greater than 225 degrees"; call it "two".

(3) Generate a key containing all the numbers in either of keys "one" or "two"; call the new key "set1".

(4) Generate a key for the condition "type equals joint"; call it "jnt".

(5) Generate a key containing all the record numbers

occurring in both keys "jnt" and "set1"; call it "jntset1" and retrieve those records.

The output from this run would be a file containing all those data records corresponding to joints with dip directions less than 45 degrees or greater than 225 degrees. The program can also be instructed to output any or all of the keys produced during the run. For example, the key for all joints could be requested, or the key for joint set 1. Use of this facility can save much computer time, as described below.

(1) Moving data records into another file can require considerable amounts of computer time.

(2) The storage space required by a key is orders of magnitude less than that required by the corresponding data records. This makes it feasible to save the results of many "retrievals" so that if the same data is required again the search operations do not need to be repeated.

(3) Keys saved from previous runs can be re-entered into the program and used to generate new keys corresponding to new combinations of conditions even without reference to the data file.

(4) Processing programs can easily be written so that they use the original data file and a key, processing only those records indicated by the key. Thus for many processing purposes it not necessary to produce a new file

of retrieved data records at any time.

2 Data Display - Contoured Orientation Diagrams

2.1 Introduction

The contoured orientation diagram is widely used in structural geology and petrofabric studies, because it has proved to be an extremely useful method of displaying three-dimensional fabric data for the purpose of inferring the fabric of the rock. A device that is so widely used and on which so many conclusions are based ought to be well understood. However, progress toward a thorough understanding of the contoured orientation diagram has been slow. An important step in this direction was taken by Kamb (1959) who pointed out some of the statistical properties of the density values obtained by conventional techniques. Another valuable contribution was made by Stauffer (1966) who investigated the characteristics of diagrams representing samples of various sizes taken from several model distributions. The present study attempts to carry this understanding a step further by means of theoretical and experimental investigation of the statistical properties of density values calculated according to the conventional definition and by two new methods described here for the first time.

Conventional methods for producing contoured orientation diagrams are reviewed in section 2. After a brief statement of some theoretical principles two new methods are introduced in section 3. Some theoretical results are presented in section 4. The new methods were compared with the conventional methods experimentally, and these experiments and their results are described in section 5. Section 6 considers the problem of estimating population parameters from contoured diagrams. Section 7 summarizes the chapter and its conclusions.

2.2 Review of Contouring Methods

A line through the origin in three-dimensional space intersects the unit sphere at two points. A two-dimensional point diagram is obtained by projecting any half of the unit sphere onto a plane. Two methods of projection are commonly used by geologists for this purpose: the stereographic projection and Lambert's equal-area projection. Conventional methods of contouring point density are based on the principle that if the centre of a circle of constant radius is placed at different locations on the sphere, the relative density of points at each location is proportional to the number of points lying inside the circle at that location. The manual methods employ a projection on which the points are plotted and counted. Since all projections involve some shape distortion these methods require a compromise between approximating the circular region on the sphere and being quick and easy to use. Stauffer (1966) gave the following summary.

"Orientation diagrams containing more than about 100 points are usually contoured for point density, and a variety of counting methods for use in contouring have been described in the literature; these are outlined below.

"(1) Schmidt Method
The number of points inside a circle (usually 1% of the total diagram area) are counted at each intersection of a squared-grid superimposed on the point diagram. For details see Billings (1954 p.113) or Turner and Weiss (1963 p.61).

"(2) Free-Counter Method
A suitable interval is determined, then the counting circle is moved over the diagram in such a way that it always contains the correct number of points for one of the contour values. Individual contour lines are drawn as the locus of the center of the moving counting circle (see Turner and Weiss 1963 p.62).

"(3) Mellis Circle Method
A 1% circle is placed over each point on the diagram and the circumference traced onto an overlay. Contour values are determined from the number of circles which overlap at any particular spot (Mellis, 1942; Turner and Weiss, 1963 p.62).

"(4) Mellis Variable-Circle Method
Mellis (1942) advocated the use of the stereographic projection for contouring, and devised a counting method in which the radius of the counting circle used is increased with distance from the center of the net; this ensures that the counting circles are of equal area throughout.

"(5) Variable-Ellipse Method
Several authors have advocated this method and a good description is given by Strand (1945 p.112). The counting cell is circular at the center of the diagram and elliptical at the periphery. This ensures that equal areas are counted over the whole net, but it applies only to equal-area projections.

"(6) Squared-Grid Method
It is also possible to count a diagram with a simple squared-grid... A counting grid designed for use on a 20-cm equal-area net contains 100 squares with sides of 0.7in (a 0.7-in square is 1% of the area of a 20-cm-diameter circle). Partial cells (rectangular and triangular) on opposite

> sides of the net are combined and each pair counted as one cell."

In addition to these graphical methods, several computer programs have been written to compute point densities using the same underlying definition. For example, Robinson (1963) described a program that calculated the intersections of planes and counted the intersections falling in predetermined non-overlapping non-circular regions of equal area. The density values were plotted by hand on a projection and contoured. Noble and Eberly (1964) described a program that calculated intersections of planes and counted the intersections falling in overlapping circular regions of equal area centred at predetermined locations. Again, the density values were plotted and contoured by hand. Muecke and Charlesworth (1966) used a program for counting poles of planes also using a circle placed at predetermined locations. This program printed the density values in the form of an equal-area projection so that only the contouring had to be done by hand.

Other programs for producing diagrams of density distributions on the sphere have been described by Baker, Wenk and Christie (1969), Warner (1969), Spencer and Clabaugh (1967), LaFountain (1970), Starkey (1969). Yet another program is described in Chapter 3 of this thesis.

2.3 Two New Contouring Methods

2.3.1 Nature of the Contouring Process

All the methods described above for producing contoured diagrams of point density imply the following definition: point density at a location on the sphere is the proportion of the sample inside a circle of specified area centred at that location. This may be written as

[2.1] $$\hat{d} = t/na$$

where $\hat{d}$ is the point density, t is the number of points inside the circle, n is the number of points in the sample and a is the area of the circle. If t is written as

[2.2] $$\sum_{i=1}^{n} w(\theta_i), \qquad w(\theta) = 1/a \quad \theta \le \theta_c$$
$$= 0 \quad \theta > \theta_c$$

where θ_i is the distance of the i^{th} point from the counting location and θ_c is the radius of the circle, then [2.1] becomes

[2.3] $$\hat{d} = (1/n) \sum_{i=1}^{n} w(\theta_i)$$

which has the form of a well known density estimator (Watson,1970; Fukunaga and Hostetler, 1973).

This definition of point density leads directly to several important concepts.

(1) If $\hat{d}$ is a density estimate, there is implied the concept of a "true" or population density d at every point on the sphere, that is, there is a density function for the population. The surface represented by the contours on a

density diagram is thus an estimate of the population density function.

(2) The "contouring" process is essentially one of smoothing the data, since density estimation using [2.3] is a smoothing operation (Thompson, 1956; Whittle, 1958; Watson, 1970). The function w is called the weighting function, and the degree of smoothing is controlled by the parameter or "effective radius" of the weighting function (Thompscn, 1956).

(3) The rectangular weighting function given in [2.2] represents only a special case of the density estimator [2.3], so that the possibility of other weighting functions is implied (Watson, 1970).

(4) The usefulness of $\hat{d}$ as an estimator of d depends on the probability distribution of $\hat{d}$.

2.3.2 The Conventional Constant-Area Method

An important step toward understanding the statistical nature of the conventional density value $\hat{d}$ defined by [2.1] was taken by Kamb (1959) who observed that, for any population, t is a binomial variable with mean np and variance np(1-p), where p is the proportion of the population inside the counting circle. Kamb noted that the coefficient of variation v of t given by

$$[2.4] \qquad v = \{np(1-p)\}^{\frac{1}{2}}/np = \{(1-p)/np\}^{\frac{1}{2}}$$

is reduced by increasing the area a of the counting circle, since this increases p (Figure 4). Reducing v was important, Kamb argued, in order to avoid values of t that differed greatly from np, and he adopted the criterion that v should not exceed one third, that is,

$$[2.5] \qquad [(1-p)/np]^{\frac{1}{2}} \leq 1/3$$

which gives

$$[2.6] \qquad p \geq 9/(n+9).$$

Since Kamb was primarily interested in detecting evidence of non-uniformity of the population, he used the relationship

$$[2.7] \qquad p = a/2\pi$$

that holds for a uniform population to obtain the following rule for determining the area a:

$$[2.8] \qquad a/2\pi \geq 9/(n+9).$$

Since the mean and variance of t for a uniform population are known for any given area a (because p is known for all locations on the sphere), the value that will be exceeded with any chosen probability can be determined. Thus the value of t for a particular counting circle provides a test of the hypothesis that the population is uniform. In this way, Kamb suggested, the contoured density diagram could provide quantitative statistical conclusions.

Two points should be noted concerning Kamb's rule. First, it was intended as a means for detecting very weak preferred orientations. Second, the distribution of t for a particular counting circle is not the same as the distribution of t_{max}, the largest value of t appearing on a diagram. When many counting circles are considered simultaneously, the probability that a given value of t will be exceeded in at least one of the circles is greater than the probability that the value will be exceeded in one particular circle. This means that if, say, the upper 5% point of the distribution of t is chosen as a critical value and preferred orientation is inferred whenever a value in excess of this critical value appears on a diagram, then the level of significance of such a test is greater than 5%, that is, more than 5% of samples drawn from a uniform population would be interpreted as having preferred orientation. The actual proportion would depend on the number, size and positions of the counting circles, and would be difficult to evaluate other than empirically.

In this study the unit of measurement of probability density is taken as the expected value of the estimate for a uniform population. Since the area of the unit hemisphere is 2π, the expected proportion of points in unit area for a uniform population is $1/2\pi$, so that the standardized conventional density estimate is

$$[2.9] \qquad \hat{d} = 2\pi t/na$$

which is a binomial variable with mean $2\pi p/a$ and standard deviation $(2\pi/a)\{p(1-p)/n\}^{\frac{1}{2}}$. Note that the weighting function w is now

$$\begin{aligned} w(\theta) &= 2\pi/a \qquad && \theta \leq \theta_c \\ &= 0 && \theta > \theta_c . \end{aligned}$$

2.3.3 The Fisher-Weighted Method

When the orientation of a bedding or joint surface is measured, the measured pole differs from the true mean pole of the surface because of (a) measurement errors and (b) local deviations of the surface from its mean orientation (Cruden, 1968). The deviation of an observed pole from the true pole is well described by a probability density function proposed by Fisher (1953) (Cruden and Charlesworth, 1966; Cruden, 1968). According to this model the probability density for an observed pole P given the true pole Q is

$$(k/4\pi \sinh k) \exp(k \cos\theta) \qquad [2.10]$$

where k is a parameter known as the concentration parameter and θ is the angle between P and Q. The density is maximum in the direction Q and is symmetric about Q. When k is large the distribution is tightly clustered about Q, while when k is small the distribution is widely dispersed. The distribution extends over the sphere, being confined to the hemisphere for practical purposes only when $k \geq 3$. The distribution is discussed more fully in Chapter 3. For repeated measurements of geological discontinuities k ranges from about 25 for very irregular surfaces to several hundred for nearly planar surfaces.

Besides the probability density for P given Q, [2.10] also gives the probability density for Q given P. Hence the probability density for a true pole Q corresponding to an observed pole P selected randomly from a sample of n observed poles $P_1, P_2, \ldots, P_n$ is

$$[2.11] \qquad (k/4\pi n \sinh k)\sum_{i=1}^{n}\exp(k\cos\theta_i)$$

where θ_i is the angle between Q and P_i and k is the same for all the P_i. Given a set of observed poles and an empirical value for k, [2.11] can be evaluated for a series of locations $Q_1, Q_2, \ldots$ and these values plotted on a projection and contoured. This empirical probability density surface embodies all the information contained in the sample, including the information about the magnitude of errors. It is an extension of the point diagram to the case in which errors are considered, and, like the point diagram, it may need smoothing before it is a useful estimator of the population density function. Smoothing is likely to be necessary when n is small or k is large. In fact, as k increases, the empirical density function [2.11] approaches a function which is zero everywhere except at the sample points, at which points it is undefined, that is, it approaches a point diagram.

A convenient way of smoothing the empirical density function [2.11] is to reduce k below its empirical value. Note that [2.11] already has the form of the density estimator [2.3] where the weighting function w is Fisher's density function [2.10]. Since [2.11] is a normalized density function, its expected value for a uniform population is the uniform density $1/2\pi$, so the standardized density estimate is

[2.12] $$\hat{d} = (k/2n \sinh k) \sum_{i=1}^{n} \exp(k \cos\theta_i).$$

The constant-area and Fisher-weighted estimators belong to a recognized class of density estimators referred to in the statistical literature as the "Parzen" class of density estimators (Wegman, 1972).

2.3.4 The Constant-Proportion Method

If Kamb's rule is used for contouring samples from populations with strong preferred orientations, the coefficient of variation of t will be far from the required value of 1/3 over most of the diagram, since over most of the sphere the population density (and therefore p) is either much smaller or much larger than that of a uniform population.

Let the required coefficient of variation be v_0. Then from [2.4] the required value p_0 of p is given by

[2.13] $$v_0^2 = (1-p_0)/np_0$$

which gives

[2.14] $$p_0 = 1/(nv_0^2 + 1).$$

If no assumptions can be made about the population density function, the only information available about p is given by the proportion t/n which is in fact an unbiased estimator of p. So we adjust the area a of the counting circle until $t/n=p_0$, and compute the standardized density estimate as before

$$\hat{d} = 2\pi t/na.$$

Note that whereas for the constant-area method a was constant and t was variable, we now have t constant and a variable. The distribution of a is complicated for non-uniform populations.

This estimator belongs to a recognized class of density estimators referred to in the statistical literature as the "k-nearest-neighbour" class of density estimators (Fukunaga and Hostetler, 1973). A similar technique for density estimation in Euclidean spaces using non-overlapping regions containing equal numbers of data points was described by Gessaman (1970).

2.4 Theoretical Investigation of Contouring Methods

2.4.1 The Constant-Area Method

Some theoretical results will be obtained for the constant-area density estimator so that these can be compared with the empirical results. This will provide a check on the empirical results and facilitate their interpretation for the other density estimators.

Since the mean of the density estimator [2.9] is $2\pi p/a$, [2.9] is an unbiased estimator of d only if

$$2\pi p/a = d,$$

that is, if the average population density over the counting circle is equal to the population density at the counting location. This is generally not true, and the bias of the estimator [2.9] can be expressed as

[2.15] $$b = \log_{10}(2\pi p/ad).$$

Since from [2.13]

[2.16] $$v^2 = (1-p)/np = (1/p - 1)/n$$

it follows that v is inversely related to p. That is, for constant population density (for example, at one counting location) v is smaller for larger counting circles, while for constant counting circle size v is smaller for larger population densities. v is also inversely related to n, so that the density estimate has a larger coefficient of variation in smaller samples.

b depends on the ratio of the average density over the counting circle to the population density at the centre of the counting circle. This ratio depends on the shape of the population density function inside the counting circle. If the population density is uniform or has a uniform slope inside the counting circle, then b will be zero. If, on the other hand, the counting circle is centred on a maximum of the population density function, the average density must be less than the maximum density, so that b will be negative. Similarly, if the counting circle is centred on a minimum of the population density function the average density will be greater than the minimum density, so b would be positive. Thus, if the population is a single Fisher distribution, we would expect negative b at the centre (maximum) of the population, zero b some distance from the centre, and probably positive b in the peripheral parts of the population. We would further expect b to depart from zero more for larger counting circles, especially at the centre and peripheral parts of the population.

In summary, we predict that as counting circle size increases at one counting location, precision will increase while accuracy decreases. With increasing distance from the centre of a density maximum (counting circle size constant), that is, decreasing population density, precision will decrease while accuracy first increases (underestimating the true density) and then decreases (overestimating the true density).

The Exact Smoothed Fisher Peak Density - Unimodal Case

If a Fisher density function with parameter k_p is smoothed using the constant-area method the smoothed density at the peak in standard units (that is, in terms of the uniform density) is the average probability density inside the circle, that is

[2.17] $$d_s = 2\pi p/a$$

where p is the probability inside the circle and a is the area of the circle. From Watson and Irving (1957)

[2.18] $$p = 1 - \exp\{k_p(\cos\theta_c - 1)\}$$

and from geometrical considerations we know that

[2.19] $$a/2\pi = 1 - \cos\theta_c$$

so that the smoothed peak density can be written

[2.20] $$d_s = (2\pi/a)\{1 - \exp(-ak_p/2\pi)\} .$$

Exact Smoothed Fisher Peak Density - Multimodal Case

Suppose a multimodal population consists of a mixture of Fisher distributions. Let the proportion of the population in Fisher distribution number 1 with parameter k_p be p_1. Then the unsmoothed peak density for that Fisher distribution is $p_1 k_p$.

Suppose this multimodal population is smoothed using the constant-area method. If p_c is the proportion of the Fisher distribution number 1 inside the circle, then the proportion of the total population inside the circle is $p_1 p_c$. Thus the smoothed peak density is now

$$d_s{*} = 2\pi p_1 p_c/a$$

or

[2.21] $$d_s{*} = (2\pi p_1/a)\{1 - \exp(-ak_1/2\pi)\}.$$

Note that the ratio of the smoothed to unsmoothed peak densities is the same as in the unimodal case.

2.4.2 The Fisher-Weighted Method

The Exact Smoothed Fisher Peak Density - Unimodal Case

If a Fisher density function with parameter k_p is smoothed using a Fisher density function with parameter k_w as the weighting function, the smoothed density at the peak is the integral over the sphere of the product of the two density functions. This product is

$$(k_p/4\pi\sinh k_p)\exp\{k_p\cos\theta\}(k_w/4\pi\sinh k_w)\exp\{k_w\cos\theta\}$$
$$= (k_pk_w/16\pi^2\sinh k_p\sinh k_w)\exp\{(k_p+k_w)\cos\theta\} .$$

Integrating over the sphere gives

$$\{k_pk_w/(k_p+k_w)\}\exp(k_p+k_w)/(8\pi\sinh k_p\sinh k_w) .$$

If we assume that $k_p \geq 5$ and $k_w \geq 5$ and ignore all terms in $\exp\{-k_p\}$ and $\exp\{-k_w\}$ this reduces to

$$(1/2\pi)\{k_pk_w/(k_p+k_w)\}$$

or, in terms of the uniform density $1/2\pi$,

[2.22] $$d_f = k_pk_w/(k_p+k_w) .$$

Thus the ratio of the smoothed to the unsmoothed peak densities is

$$k_w/(k_p+k_w) .$$

The Exact Smoothed Fisher Peak Density - Multimodal Case

The density function for the Fisher distribution number 1 is the normalized Fisher distribution reduced by the factor F_1. Thus the integral for the smoothed peak density (ignoring the influence of the other Fisher distributions) is changed by the factor p_1; hence the smoothed peak density is

[2.23] $$d_f^* = p_1 k_p k_w/(k_p+k_w).$$

Again the ratio of the smoothed to the unsmoothed peak densities is the same as in the unimodal case.

2.4.3 The Constant-Proportion Method

The Exact Smoothed Fisher Peak Density - Unimodal Case

If a Fisher distribution with parameter k_p is smoothed using the constant-proportion method, the smoothed peak density is again $2\pi p/a$ where a is the area of the circle that contains the proportion p of the population. In this case we express this in terms of p rather than in terms of a. From [2.18] and [2.19] above we have

$$a = 2\pi(1-\cos\theta_c)$$

and

$$\cos\theta_c = 1 + \ln(1-p)/k_p$$

so that the smoothed peak density is

$$d_v = 2\pi p/\{2\pi(-\ln(1-p)/k_p)\}$$

or

[2.24] $$d_v = -pk_p/\ln(1-p).$$

Note that the ratio of the smoothed to the unsmoothed peak densities in this case is

$$-p/\ln(1-p)$$

which is independent of k_p.

The Exact Smoothed Fisher Peak Density - Multimodal Case

This method requires that the proportion of the total population in the circle be p_o, that is,

$$d_v^* = 2\pi p_o/a_o$$

where a_o is the area of the circle that fulfills the above condition. Now $p_o = p_1 p_c$, so that $p_c = p_o/p_1$. Hence from [2.18] and [2.19] above

$$a = -2\pi \ln(1 - p_o/p_1)/k_p$$

so that

[2.25] $$d_v^* = -p_o k_p / \ln(1 - p_o/p_1).$$

Note that for this method the ratio of the smoothed to unsmoothed peak densities is not the same as in the unimodal case; however, the ratio is still independent of k_p.

2.5 Experimental Investigation of Contouring Methods

2.5.1 The Experiments

Let (θ,ϕ) be spherical coordinates related to rectangular coordinates (x,y,z) by

[2.26] $$x = \sin\theta\cos\phi$$

[2.27] $$y = \sin\theta\sin\phi$$

[2.28] $$z = \cos\theta.$$

Random samples from a Fisher distribution were generated using the algorithm given by Stauffer (1966):

[2.29] $$\cos\theta = (1/k)\log\{(1-r)\exp(k) + r\exp(-k)\}.$$

Replacing r by numbers drawn randomly from a uniform distribution on the interval (0,1) and combining the resulting values of θ with values of ϕ drawn from a uniform distribution on $(0,2\pi)$, will give a series of points (θ,ϕ) whose distribution is the Fisher distribution with its polar axis the z-axis. This was accomplished using the program SAMPLEGEN.

For each set of conditions, that is, given values of the population concentration parameter k_p and the sample size n, 50 samples were generated. Using a series of counting locations at distances γ from the centre of the population of 0, 5, 10, . . . degrees up to 25 or 35 degrees depending on the shape of the population, the population density function was estimated at each counting location using the three different estimators with various parameter values. This procedure was followed using a population parameter k_p of 50 and a sample size n of 100. The experiment was repeated using the same population and two other sample sizes, n=50 and n=200, and again using n=100 and k_p=100.

The estimators used were the constant-area estimator with $a/2\pi$ =1%, and in some cases with $a/2\pi$ =2%, 5% and 10%; the constant-area estimator with the counting-circle size determined from Kamb's rule; the constant-proportion estimator with v_o equal to 1/3, 1/5, and in some cases 1/7, 1/10 and 1/20; and the Fisher-weighted estimator with the parameter k_w equal to 25, 50, 100, 242, and in some cases 500.

For each experiment, each estimator type and parameter value, and each counting location, the 50 density estimates $\hat{d}_1, \hat{d}_2, \ldots, \hat{d}_{50}$ were used to compute the standard deviation of the estimates

$$[\{\Sigma \hat{d}_i^2 - (\Sigma \hat{d}_i)^2/50\}/49]$$

the mean of the estimates

$$\Sigma \hat{d}_i/50$$

the coefficient of variation of the estimates

$$[2.30] \quad \hat{v} = [\{\Sigma \hat{d}_i^2 - (\Sigma \hat{d}_i)^2/50\}/49]\ (50)/\Sigma \hat{d}_i$$

and an estimate $\hat{b}$ of b given by

$$[2.31] \quad \hat{b} = \log_{10}(\Sigma \hat{d}_i/50d).$$

d is found for each population and each counting location by evaluating the density function [2.10] for $k=k_p$ and $\theta=0$, 5, . . . degrees.

2.5.2 Results and Discussion

Results for the experiment n=100, $k_p=50$ will be discussed before comparisons are made with other experiments. The population density function

$$\{k/(4\pi \sinh k)\} \exp\{k \cos\theta\}$$

is illustrated in Figure 5. It can be seen that the density decreases to the uniform density $(1/2\pi)$ at about $\theta=23$ degrees.

The points representing one estimator type and one parameter (different counting locations) lie in a more or less smooth curve, the properties changing gradually from the centre of the population to the periphery (Figures 6, 7 and 8). Each such curve will be referred to as a "parameter curve". The points representing one estimator type and one location (different parameters) also lie on a smooth curve, the properties changing gradually as the parameter changes (Figures 9, 10 and 11). Each such curve will be referred to as a "location curve".

We compare the empirical results for the constant-area estimator with the theoretical results for that estimator. The location curves (Figure 9) show that as counting circle size increases, coefficient of variation decreases and bias increases. Both relationships are in accord with the theoretical results for this estimator. The parameter curves for the same estimator (Figure 6) show that with increasing distance from the centre of the population, coefficient of variation increases while bias changes from negative to positive. These results were also predicted. Notice in particular how the coefficient of variation of the constant-area estimator with $a/2\pi=1\%$ becomes extremely large with decreasing population density, while its bias remains small. In contrast, the coefficient of variation of the estimator with $a/2\pi=8.3\%$ (Kamb's rule) remains very small while its bias changes from strongly negative to strongly positive with increasing γ. The parameter curves for the Fisher-weighted estimator (Figure 8) are similar to those for the constant-area estimator (Figure 6). This means that similar combinations of properties (at different locations) result from large and small k_w as are obtained using the constant-area estimator with respectively small and large counting circles. However, the Fisher-weighted location curves for large γ lie slightly closer to the origin than the corresponding curves for the constant-area estimator (Figures 12 and 13). This difference disappears for $\gamma = 15$ degrees (Figure 14), and there is no consistent difference for smaller γ (Figures 15, 16 and 17). Thus, in the peripheral parts of the population, the Fisher-weighted estimator provides a slightly more advantageous compromise between coefficient of variation and bias than is possible with the constant-area estimator.

The parameter curves for the constant-proportion estimator (Figure 7) are quite different from those of either of the other two estimators (Figures 6 and 8). This estimator combines the characteristics of a small-circle constant-area estimator at high population densities with those of a large-circle constant-area estimator at low population densities. Thus it avoids the large coefficients of variation exhibited by the small-circle constant-area estimators at low population densities without incurring inordinately large smoothing effect and hence negative bias characteristic of the large-circle constant-area estimators at high population densities. The location curves for this estimator lie very close to those for the other two estimators (Figures 12 to 17).

Although the constant-area estimator is designed to have a constant coefficient of variation, the parameter curves for this estimator (Figure 7) show that the coefficient of variation is not actually constant. The reasons for this are as follows. The random variable $\hat{d}$ is given by

$$\hat{d} = 2\pi t/na.$$

Now t is a binomial variable with mean and variance depending on n and p. But p changes from sample to sample (because the area a of the counting circle changes from sample to sample). That is, both a and p are also variables. Thus the theoretical distribution of $\hat{d}$ is complicated, especially since the relationship between a and p is complicated. p is in fact the integral of the population density function over the counting circle; the way that this changes with a depends on the value of the function at the counting location and the shape of the function inside the counting circle.

It appears that there is a limit to how small v can be made, especially at the centre of the population where v appears to have a lower limit of about 0.16 for k_p=50 and n=100.

Comparison of the results for k_p=50, n=100 with the results for other sample sizes reveals that the general relationships described above still hold. The location curves lie closer to the origin when $n=200$ and farther from the origin when $n=50$ (Figures 18, 19 and 20), confirming the intuitive expectation that more advantageous combinations of coefficient of variation and bias are possible with larger samples. At the centre of the population the effect of increasing n on the constant-area estimator and Fisher-weighted estimator with given parameters is to increase precision without changing the accuracy (Figures 18 and 20), while for the constant-proportion estimator increasing n increases both accuracy and precision (Figure 19). This is because for the constant-area estimator and the Fisher-weighted estimator the degree of smoothing and therefore the accuracy are fixed because the parameter of the weighting function is fixed, whereas the degree of smoothing for the constant-proportion estimator decreases as n increases as a result of the imposed relationship between its parameter v_0 and n.

Figure 18B showing how the properties of the constant-area estimator at $\gamma=25$ degrees change with sample size at first sight appears to indicate that as sample size increases the mean of the estimator decreases and then increases again. The explanation of this figure is as follows. Each estimate of the density can be written as

$$(2\pi/a)(\Sigma t_i/50n) \tag{2.32}$$

where t_i is the number of points out of the i^{th} sample that fall inside the counting circle. This estimate is binomial, mean $2\pi p/a$, standard deviation $(2\pi/a)\{p(1-p)/50n\}^{\frac{1}{2}}$. Using

$a/2\pi=1\%$, $n=50$, 100 and 200, and assuming $p=0.008$ (based on the estimated densities of 0.9, 0.6 and 0.8; see Appendix 3), approximate standard deviations of the means for the three estimates of the density for $a/2\pi=1\%$ were computed. These standard deviations are 0.20, 0.14 and 0.10 for $n=50$, $n=100$ and $n=200$ respectively. The intervals formed by the estimates plus and minus three standard deviations, transformed by dividing by the true density and taking the base 10 logarithm, were drawn on Figure 18B. At once it is seen that the changes in the estimated density as n increases are well within experimental error, and therefore not significant. The curves for $a/2\pi=2\%$ and $a/2\pi=5\%$ and the curves for the Fisher-weighted estimator in Figure 20B follow the same pattern because all the estimates were computed from the same random samples for any given sample size. That is, because the set of 50 samples for $n=50$, say, happened to contain more than the expected number of points in the vicinity of the counting location at $\gamma=25$ degrees, all of the estimates from that counting location for $n=50$ reflect this.

Figures 21 and 22 show the effect at the centre of the population of changing the population parameter k_p. The estimators with fixed weighting functions show an increase in precision and decrease in accuracy with increasing k_p. The decrease in accuracy is due to the fact that the sharper population peak for $k_p=100$ is smoothed relatively more than the flatter population peak for $k_p=50$ by the fixed-weighting-function estimators. However, the change in accuracy is very slight for the constant-proportion estimator (Figure 22). This implies that the ratio of the mean of the estimator to the true peak density of the population is independent of the true peak density - a result we have already obtained theoretically. This suggests that the estimator might be used to estimate k_p when k_p is unknown. Figure 19 confirms, however, that this ratio is not independent of n.

2.6 Use of the Orientation Diagram

2.6.1 The Modelling Process

The model-building process is explained by, for example, Box and Jenkins (1970) in the context of time-series analysis. Their illustration is reproduced in Figure 23. The essential steps in the model-building process are selection of a model type, estimation of parameters, and diagnostic checking to decide whether the model is adequate. In modelling fabrics, the orientation diagram provides a basis for selection of a model type (that is, the number and type of fabric elements present), and a means for estimating parameters, especially location parameters. It can also be used to estimate concentration parameters, and this function is especially important if the fabric elements cannot be satisfactorily isolated by subdividing the sample. Methods for checking the adequacy of the preliminary model obtained from the diagram are still lacking, and statistically sound models for fabric data cannot be constructed until such techniques become available.

2.6.2 Estimation of Fisher Concentration Parameters

Here we concentrate on the role of the orientation diagram as a parameter estimation tool; we suggest that the information obtained about the different density estimators in this study makes it possible to obtain estimates of concentration parameters of Fisher-distributed point clusters from orientation diagrams making maximum use of the information available.

The problem and possible solutions may be explained as follows. For a population whose density function is a single Fisher distribution the concentration parameter may be related in a simple way to the peak density (a measure of size) or to the half-maximum radius (the distance from the peak at which the density is half the peak density; a measure of shape).

Consider now a multimodal population whose density function is the sum of several Fisher distributions which, since the whole density function integrates to unity, are now scaled down, each by a different factor depending on the proportion of the population lying in each point cluster.

For each point cluster, the relationship between concentration parameter and peak density is now complicated by the addition of the scaling factor. However, note that the half-maximum radius is not changed by scaling.

Finally, since the empirical density function (the density diagram) is produced by a smoothing operation, we must consider the effect of this smoothing. Its effect on the peak density is known exactly from the equations derived in section 2.4. Its effect on the measure of shape is unknown, although if it is assumed that the smoothed Fisher distribution has itself the shape of a Fisher distribution, then the effect of smoothing on shape may be inferred from the known effect on peak density; this assumption remains to be verified, however.

Thus the empirical density function for a particular mode represents a Fisher distribution with unknown parameter k_p, not only smoothed but also scaled down by an unknown factor p_1. Two possible ways of approaching the problem of determining the unknown parameter k_p are suggested.

(1) The relative proportions of a population in different regions of the sphere are estimated by the relative numbers of sample points in those regions. Thus, once the ranges of the different Fisher distributions have been estimated from a contoured diagram, then the proportion of the population in a particular point cluster can be determined by counting points on a point diagram. In the

case of large samples it may be more feasible to estimate this proportion from the contoured diagram itself. This can be done by either (a) measuring areas enclosed by contours with a planimeter, multiplying by appropriate density values and summing, or (b) simply summing the density values on the diagram, provided they are uniformly spaced. Once the proportion p_1 has been estimated, the concentration parameter of the Fisher distribution can be determined from the peak value on the diagram using the relationships [2.21], [2.23] or [2.25].

(2) A different way of obtaining the unscaled smoothed peak density uses only the mode or peak being estimated, not involving the rest of the diagram at all. It depends on the assumption that the smoothed Fisher distribution has itself the shape of a Fisher distribution. In other words, it is assumed that smoothing a Fisher distribution simply changes it into another Fisher distribution with smaller parameter.

This method is based also on the observation that the radius at which a Fisher density function has half the peak density depends only on k and is not changed by scaling. The relationship is, for $k \geq 5$,

$$k = (\ln 0.5)/(\cos\theta_{\frac{1}{2}} - 1)$$

where k is the parameter of the distribution and $\theta_{\frac{1}{2}}$ is the half-maximum radius. That is, the "half-maximum radius" of a scaled Fisher distribution provides an immediate solution for its concentration parameter, regardless of the scaling.

Thus, assuming that the smoothed distribution has the shape of a Fisher distribution, the half-maximum radius reveals the parameter of the smoothed distribution; this is then translated into the smoothed peak density, which is related to the concentration parameter of the unsmoothed Fisher distribution in a known way for a given estimator.

2.6.3 Optimization of Density Estimators

Optimization of density estimators (or smoothing methods) has been discussed in various contexts. For example, Fukunaga and Hostetler (1973) have considered the optimization of constant-proportion estimators. Optimization of estimators of the constant-area and Fisher-weighted type has been studied by Watson and Leadbetter (1962), Cacoullos (1966) and Epanechnikov (1969). Most workers have used as an optimization criterion minimization of the mean integrated squared error (Whittle, 1958; Elkins, 1968; Fukunaga and Hostetler, 1973). This is the integral over the domain of the distribution of the local mean squared error, which is the sum of the variance of the estimator and the square of the bias, where bias is the difference between the true density and the mean of the estimator.

The problem with this approach is that the estimator that minimizes the mean integrated squared error depends on the nature of the distribution being estimated, and the rules for optimizing the estimator can be defined only for certain classes of distributions. In any case, for our purposes it is not useful to mix variance and bias, where bias measures deviation from the true density; as long as we can estimate the smoothed density we can relate this to the true density. Some empirical optimization criteria have been suggested for the constant-proportion estimators, for example, Loftsgaarden and Quesenberry (1965) indicated that a proportion of $n^{-\frac{1}{2}}$ gave good results.

Thompson (1956), in a discussion of optimum smoothing of two-dimensional fields, pointed out that the optimum degree of smoothing depends on the ratio of the scales of "signal" and "noise". (In the case of estimating a density function, the "signal" is the true density function and the "noise" is the random variation in the empirical density function due to the finite size of the sample.) For a true density function consisting of a mixture of Fisher distributions, the scale of the "signal" may be thought of as the distance between the centres of the Fisher distributions, while the scale of the "noise" depends on the density of sample points. Thus the scale of the "noise" varies over the sphere, suggesting that the degree of smoothing (that is, the effective radius of the smoothing function) should be varied according to the density of sample points, as it is in the constant-proportion method.

In the geological literature, in connection with contoured orientation diagrams, some rules have been suggested for optimizing the constant-area estimator. Flinn (1958) recommended the rule $a/2\pi=100/n$. Starkey (1975) has reiterated this suggestion. We have already described Kamb's rule for nearly uniform populations.

An optimization criterion must be formulated with reference to a specific objective. The first function of the orientation diagram is to suggest the number of fabric elements present. Now a diagram prepared from a sample of n points may show any number of maxima from 0 to n inclusive depending on the degree of smoothing; in the absence of any prior knowledge of the population, no particular number of maxima is any more likely to be correct than any other. It is suggested that the greatest amount of information is available from a series of diagrams using different degrees of smoothing. Consideration of the shapes of the various peaks of the density function at different degrees of smoothing in conjunction with an assumption about the shapes likely to be found in the population, for example, Fisher distributions, can then lead to selection of a tentative model type. Whether any type of estimator is "better" in any sense than the others for this purpose is not apparent from the data available.

The second function of the orientation diagram in the modelling process is to provide estimates of parameters. For determining location or peak density of a point cluster we require an estimator that combines an acceptable coefficient of variation at the peak with the smallest possible effective radius, so that the distorting influence of neighbouring point clusters is as small as possible. If the degree of underestimation of the peak density of a unimodal population is taken as a measure of the effective radius, the fact that the location curves at the population peak for the constant-area and Fisher-weighted estimators are essentially identical may be taken as indicating that these two estimators are equally good for this purpose. Since the constant-proportion estimator cannot be characterized by a constant effective radius that can be extrapolated to multimodal situations, it is not apparent from the available data how it should be regarded in this context.

The amount of smoothing to be used in producing an orientation diagram should perhaps depend in part on the precision of the observations. The smoothing could then be thought of as being of two kinds: first, smoothing in order to represent the amount of uncertainty in the data, and, second, additional smoothing (if required) in order to estimate the population. Note that the uncertainty in the data refers to all sources of error whose scale is smaller than that of the investigation. For example, suppose the orientation of a joint surface is measured. That measurement can be used to represent (a) the orientation of that part of the joint surface with which the measuring device was in contact, (b) the orientation of the entire joint surface, or the orientation of a joint set (c) in a small part of an outcrop, (d) over a whole outcrop, or (e) over a whole region. In all these cases, the error in the measurement includes the error in reading the scale of the instrument, and the error in aligning the instrument with that part of the joint surface with which it was in contact. In case (b) an additional error is the deviation of the joint surface at the point of contact from the mean orientation of the joint surface. In cases (c), (d) and (e) a fourth error that must be recognized is the deviation of that particular joint from the mean of the joint set in the small part of the outcrop, the whole outcrop, or the region (assuming such a mean exists). The probable size of this error increases with the size of the area being represented.

2.7 Summary and Conclusions

Conventional methods of producing contoured orientation diagrams may be seen to correspond to a special case of the well-known density estimator

$$\hat{d} = (1/n) \sum_{i=1}^{n} w(\theta_i)$$

where n is the sample size, θ_i is the distance of the i^{th} point from the counting location and w is a weighting function. The special case is that in which the weighting function w is the rectangular function

$$w(\theta) = 1/a \qquad \theta \le \theta_c$$
$$= 0 \qquad \theta > \theta_c$$

where a is the area of the counting circle and θ_c is the radius of the counting circle. The conventional methods are collectively referred to as the constant-area method.

Kamb (1959) suggested that the area of the counting circle should be chosen so that the coefficient of variation of the density estimate does not exceed one third. This requires that a specified proportion of the population lie within the counting circle, a condition which can be achieved exactly only for uniform populations. It is here suggested that this condition can be approximated for any population by choosing the size of the counting circle at each counting location so that the specified proportion of the sample lies within the counting circle. This is called the constant-proportion method.

Replacing the square weighting function by Fisher's probability density function provides a density estimate that is algebraically identical to the result of incorporating errors that are distributed about their mean according to Fisher's distribution. This is called the Fisher-weighted method.

Theoretical investigation of the three methods for unimodal Fisher-distributed populations yields a relationship between the parameter of the estimator, the true peak density of the population and the smoothed peak density (mean of the estimator) for each estimator type. This permits the true peak density of a unimodal population to be determined from the empirical peak density. For populations consisting of a mixture of Fisher distributions a fourth variable is added to each equation: the proportion of the population in a particular point cluster. In this case, an estimate of this proportion is required as well as the empirical peak density.

Experimental investigation of the three methods using unimodal populations reveals that the form of the weighting function is less important in determining the properties of the estimator than whether the parameter of the weighting function is determined a priori or a posteriori at a particular counting location. Thus the properties of the constant-area estimator and the Fisher-weighted estimator tend to change in a similar manner with increasing distance from the population peak while those of the constant-proportion estimator change in a different way.

The process of inferring the subfabric for a particular fabric element from an orientation diagram is seen as a model-building process consisting of model-selection, parameter-estimation, and adequacy-testing steps. While the orientation diagram is useful for model selection and parameter estimation, testing procedures are still lacking. The modelling process is thus incomplete, and statistically sound models cannot be obtained from the orientation diagram alone.

For each of the estimator types studied equations have been derived relating the smoothed peak density of a Fisher-distributed point cluster to the unsmoothed peak density and hence to the concentration parameter. For multimodal populations, an additional complication is the scaling that results from calculating the density estimates using the total sample size rather than the number of points in one point cluster. Two ways of dealing with this scaling are suggested: (a) determine the scaling factor by counting or summing over the range of the point cluster, and then "unscale" the peak being estimated, and (b) ignore the scaling and obtain the unscaled peak density by means of a measure of shape which is a function of the unscaled peak density but is independent of scaling. This requires the assumption that the smoothed Fisher distribution has itself the shape of a Fisher distribution. This second method may have an advantage when distributions overlap, since the half-maximum radius can be measured using those parts of the distribution that are free from interference.

Various rules and optimization criteria that have been suggested for determining the degree of smoothing to be applied to the data either are arbitrary or depend on the nature of the unknown population. For the purpose of model-type selection, no criterion can be defended for choosing one degree of smoothing over another. Different fabric elements may be most accurately represented by different degrees of smoothing. It is suggested that the procedure is best performed using a series of diagrams produced by various degrees of smoothing. There is no evidence that any one of the estimator types is to be preferred for this purpose.

For maximum efficiency of parameter estimation, a density estimator is required that provides an acceptable balance between the coefficient of variation of the estimator at the peak being estimated and the distorting influence of neighbouring peaks. The constant-area and Fisher-weighted estimators appear equally suitable; the properties of the constant-proportion estimator with the unimodal populations used in this study cannot be extrapolated to multimodal populations.

3 Numerical Characterization

3.1 Introduction

The subject of this chapter is the characterization of single fabric elements by means of statistics that not only describe the fabric element but also can be used in statistical tests. There are two kinds of orientation data: axes, which can be thought of as diameters of the unit sphere, and vectors, which can be thought of as radii of the unit sphere. There are two kinds of descriptive statistics that are important for orientation data. Both kinds include measures of mean and dispersion. For reasons that will be explained later, the first kind are suitable for characterization of vector data and will be referred to as vector statistics, while the second kind are suitable for characterization of axis data and will be referred to as axis statistics. The axis statistics include the direction of minimum density as well as the direction of maximum density or mean direction.

Statistical tests are based on the properties of a theoretical distribution model, the parameters of which are estimated from the data. Fisher (1953) introduced a model suitable for representing clusters of vectors. The parameters of the distribution can be estimated from the vector statistics, so that those statistics can be used to perform tests on vector data. Additional testing procedures have been derived by Watson (1956a, 1956b), Watson and Williams (1956), Watson and Irving (1957) and Stephens (1967, 1969). Barton (1974) described a graphical procedure for obtaining the vector statistics.

While most fabric data are axes, some can be vectors. For example, the normals to bedding surfaces can be vectors if the facing direction is known and taken into account. Cruden and Charlesworth (1966) used the vector statistics and Fisher's distribution to characterize and test such data. Watson (1960, 1965) developed some tests for the case of samples of vectors drawn from several Fisher distributions with coplanar means, in response to the need for methods of treating observations of cylindrically folded bedding. This application has been further discussed by Cruden (1968) and Cruden and Charlesworth (1972).

Bingham (1964) studied a model suitable for representing a cluster of axes. The parameters of this distribution can be estimated from the axis statistics, so that these statistics can be used in tests for axis data. A special case of Bingham's distribution has been studied by Dimroth (1962, 1963) and Watson (1965). This special case has axial symmetry and is known as the Dimroth-Watson distribution. Two other models of axially symmetric great-circle distributions were studied by Selby (1964), but they have not been developed.

While the axis statistics and Bingham's distribution are suitable for characterizing and testing axis data, their development has been relatively recent and little use has been made of them. The axis statistics were used by Fara and Scheidegger (1963) and Scheidegger (1964) in the analysis of earthquake fault planes, and by Scheidegger (1965) for characterizing clustered axes such as sedimentary grain axes, but no testing procedures were available at that time. Use of these statistics to characterize the distributions of long axes of till stones has been discussed by Ramsden (1970) and Mark (1973), but again testing procedures based on Bingham's distribution were not used.

Both these models are unimodal, so that only unimodal distributions can be tested. None of the statistics mentioned above are adequate to describe multimodal distributions, in which each mode must be characterized separately. A statistical test is valid only if the model on which it is based is a reasonably good approximation of the actual distribution. In all uses of the vector and axis statistics and tests based on the corresponding distribution models, the adequacy of the model must be established. This is done by combinations of parametric tests, non-parametric tests and data displays.

Fisher's model can be used to obtain meaningful results with axis data by assigning a sense or direction to the axes provided that

(1) the axes have a unimodal axially symmetric

distribution, as indicated by an adequate fit to Fisher's model,

(2) the dispersion of the axes is sufficiently small (estimated concentration parameter of Fisher's model > 3, which means that $Pr(\theta > \pi/2) \approx 0.05$), and

(3) sense is assigned to the axes so that the resulting vectors form a single cluster.

The vector statistics and tests based on Fisher's model have been used in the analysis of the long axes of till stones (Andrews and Shimizu, 1966; Andrews and King, 1968; Cowan, 1968; Andrews and Smith, 1966,1970; Andrews, 1971). In these papers points (2) and (3) were considered (the treatment of condition (3) was greatly improved by Mark (1971)) but the fit of the data to Fisher's distribution was not tested. This amounts to an implicit assumption that the long axes analyzed had unimodal axially symmetric distributions. In view of the well-known tendency of the long axes of till stones to display multimodal distributions (Holmes, 1941; Glenn, Donner and West, 1957; Harrison, 1957; Kauranne, 1960; Lindsay, 1970) results based on such an assumption are very unlikely to be accurate. In these studies a test of uniformity based on Fisher's model was applied to the axis data; the test is not under any circumstances valid for axis data. These shortcomings have been discussed in detail by Ramsden (1970) and Mark (1973).

The vector and axis statistics are described in section 3.2, and the important distribution models in section 3.3. Tests for the adequacy of the different models and the use of these tests as means for characterizing a distribution of unknown form are described in section 3.4. In some applications it is desirable to weight observations; however, in the literature cited above all observations are assumed to carry equal weight. Therefore, some procedures for weighted data are suggested in section 3.5. Computer programs to implement the methods recommended in this chapter are described in section 3.6. Section 3.7 contains some examples, and section 3.8 summarizes the recommended procedures.

3.2 Descriptive Statistics

3.2.1 Distributions on the Sphere

The most commonly encountered spherical point distributions are illustrated in Figure 24. A radius of the unit sphere (unit vector) cuts the unit sphere at one point. A cluster of vectors thus gives rise to a polar distribution (Figure 24A). A diameter of the unit sphere (axis) cuts the unit sphere at two points, so that a cluster of axes gives rise to a bipolar distribution (Figure 24B). Axes scattered about a plane yield an equatorial or girdle distribution (Figure 24C). Distributions A, B and C may be axially symmetric or asymmetric. Axes or vectors with no preferred orientation give rise to a uniform distribution (Figure 24D). These are the simple distributions; distributions of greater complexity can be regarded as mixtures of these simple distributions. For example, vectors clustered about two directions produce a mixture of two polar distributions, while axes clustered about two preferred orientations yield a mixture of two bipolar distributions.

3.2.2 Vector Statistics

Let $A_i=[x_i, y_i, z_i]'$, $i=1,...,n$ be a sample from a population of unit vectors. A useful measure of the direction of maximum concentration is the vector of means

$$[3.1] \quad M = [\Sigma x_i/n, \Sigma y_i/n, \Sigma z_i/n]' = \Sigma A_i/n .$$

Let $m=|M|$. Note that $0 \le m \le 1$, and that if all the A_i are equal to A then $M=A$ and $m=1$. So m is a measure of concentration.

3.2.3 Axis Statistics

Let A_i, $i=1,...,n$, represent a sample from a population of axes. Since each axis could be equally well represented by $-A_i$, measures are required in which A_i and $-A_i$ are equivalent. Such measures are provided by the matrix of variances and covariances

$$[3.2] \qquad T = (1/n)\begin{bmatrix} \Sigma x_i^2 & \Sigma x_i y_i & \Sigma x_i z_i \\ \Sigma x_i y_i & \Sigma y_i^2 & \Sigma y_i z_i \\ \Sigma x_i z_i & \Sigma y_i z_i & \Sigma z_i^2 \end{bmatrix} = A_i A_i'/n.$$

T has three eigenvectors M_1, M_2 and M_3 which identify the axes of minimum, intermediate and maximum concentration, and three eigenvalues m_1, m_2 and m_3 which measure the concentrations of the sample along those three axes. Note that trace(T)=1, so that $m_1+m_2+m_3=1$, and that $0 \le m_i \le 1$, i=1,2,3; also that if all the $A_i=A$, then $T=AA'$, $M_3=A$ and $m_3=1$ while M_1 and M_2 are indefinite and $m_1=m_2=0$. Note further that if the A_i are not all equal and are coplanar, M_1 is normal to the A_i and $m_1=0$, while M_2, M_3, m_2 and m_3 are determined by the distribution of the A_i in the plane. For further discussion of the interpretation of these statistics see Watson (1966) and Mardia (1972).

3.3 Distribution Models

3.3.1 Fisher's Distribution

According to this model the probability is distributed with axial symmetry about a polar vector U so that the density is maximum at U and minimum at -U. The density in a given direction A is proportional to

$$[3.3] \qquad \exp(kU'A)$$

where k is a constant. The larger k the more the probability is concentrated about U; hence k is called the concentration parameter.

If A_i, i=1,...,n, is a sample from a Fisher distribution, the maximum likelihood estimate of the polar vector is the mean vector M defined in section 3.2.2 (Mardia, 1972, p.250). Furthermore, an unbiased estimate of k when k is large is

[3.4] $(1-2/n)/(1-m)$ (Mardia, 1972, p.251).

If n is also large, the variance of this estimate is approximately

[3.5] k^2/n (Mardia 1972, p.250).

The semi-apical angle of the $(1-\alpha)$ confidence cone for the polar vector is

[3.6] $\arccos\{1 - F_0(1-m)/m(n-1)\}$

where F_0 is the upper $100\alpha\%$ point of the F-distribution with 2 and 2n-2 degrees of freedom (Mardia, 1972, p.261).

There are three drawbacks to using Fisher's distribution and the vector statistics with axes.

(1) For representing axes, the vector A is equivalent to the vector -A . However, in computing the vector statistics these two vectors are not equivalent, so the vectors used to represent the axes cannot be chosen arbitrarily. They must be chosen so they form a single cluster on the sphere.

(2) Use of Fisher's distribution cannot be justified

when observed axes have a large dispersion, that is, when $k<3$, since Fisher's distribution cannot then be used as a distribution on a hemisphere. This may be shown by setting $k=3$ and $\theta = \pi/2$ in [3.17]. This gives the probability that an angle θ will be observed that is greater than $\pi/2$ when $k=3$, which is

$$\exp(-3) = 0.05.$$

That is, less than 95% of the probability is within one hemisphere when $k<3$.

(3) The parametric test for uniformity based on Fisher's distribution described in section 3.4.2 is not valid for axes.

These drawbacks are overcome by using the axis statistics and Bingham's distribution described in the next section.

3.3.2 Bingham's Distribution

According to this model the probability is distributed with respect to three orthogonal axes U, V and W so that the density in a given direction A is proportional to

$$[3.7] \qquad \exp(k_1(U'A)^2 + k_2(V'A)^2 + k_3(W'A)^2)$$

where k_1, k_2 and k_3 are constants. Since the sum of the k's is arbitrary k_3 is taken as zero (Mardia, 1972, p.235). Bingham's model contains several important special cases (Mardia, 1972, p.236): (a) For $k_1=k_2=k_3$ it reduces to the uniform distribution. (b) For $k_2=k_3=0$ it has axial symmetry. (c) For $k_1 >> k_2 > k_3 = 0$ it is an asymmetric bipolar distribution, while for $k_1 << k_2 < k_3 = 0$ it is an asymmetric girdle distribution.

If A_1, i=1,...,n, represents a sample from a Bingham distribution, the maximum likelihood estimates of the principal axes are the eigenvectors M_1, M_2 and M_2 of the matrix T defined in section 3.2.3 (Mardia, 1972, pp.254-256). The maximum likelihood estimates of k_1 and k_2 can be obtained from tables or nomograms in Bingham (1964) using the eigenvalues m_1, m_2 and m_3, but the procedure is awkward. When the distribution is nearly axially symmetric estimates of k_1 and k_2 can be calculated using an approximation given by Mardia (1972, p.278).

The special case in which the distribution has axial symmetry is so important that it has been studied as a separate distribution known as the Dimroth-Watson distribution.

3.3.3 Dimroth-Watson's Distribution

According to this model (Mardia, 1972) the probability is distributed with axial symmetry about a polar axis U so that the density in a given direction A is proportional to

$$\exp\{k(U'A)^2\} \qquad [3.8]$$

where k is a constant again called the concentration parameter. The model differs from Fisher's in having a plane of symmetry normal to U. For k=0 it reduces to the uniform distribution, while for $k>0$ it is a bipolar distribution with the probability concentrated about U and -U.

If A_1, $i=1,\ldots,n$, represents a sample from a Dimroth-Watson distribution with $k>0$, the maximum likelihood estimate of the polar axis is given by the eigenvector M_3 associated with the largest eigenvalue m_3 of the matrix T defined in section 3.2.3 (Mardia, 1972, p. 253). The maximum likelihood estimate of k can be obtained from tables in Mardia (1972) using m_3. Mardia (1972) has shown that the $(1-\alpha)$ confidence cone for the polar axis is given by

$$(m_1\cos^2\phi + m_2\sin^2\phi)\sin^2\theta + m_3\cos^2\theta \geq c \qquad [3.9]$$

where $c=m_3(n-2-2F_0)/(n-2)$ and F_0 is the upper $100\alpha\%$ point of the F-distribution with 2 and n-2 degrees of freedom, and (θ,ϕ) are spherical coordinates referred to M_1, M_2 and M_3 as axes. Replacing the inequality by an equality and setting $\phi=0$ gives the radius on the cone in the plane normal to M_2:

$$\theta = \arccos\{(c-m_1)/(m_3-m_1)\}^{\frac{1}{2}}. \qquad [3.10]$$

Setting $\phi=\pi/2$ gives the radius in the plane normal to M_1:

$$\theta = \arccos\{(c-m_2)/(m_3-m_2)\}^{\frac{1}{2}}. \qquad [3.11]$$

For $k<0$ the Dimroth-Watson model represents an equatorial distribution with the probability concentrated about the plane normal to the polar axis. The maximum likelihood estimate of the polar axis in this case is M_1 (Mardia, 1972, p. 253) and that of k can again be obtained from tables in Mardia (1972), this time using m_1. If (θ,ϕ)

are spherical coordinates referred now to the axes M_2, M_3, M_1, Watson (1965) showed that the 1-α confidence cone for the polar axis is defined by

[3.12] $$m_1\cos^2\theta + (m_2\cos^2\phi + m_3\sin^2\phi)\,\sin^2\theta \le c$$

where $c=m_1(n-2-2F_o)/(n-2)$ and F_o is the upper 100α% point of the F-distribution with 2 and n-2 degrees of freedom. As before, we compute the radius of the cone in the plane normal to M_3 by changing the inequality to an equality and setting $\phi=0$:

[3.13] $$\theta = \arccos\{(m_2-c)/(m_2-m_1)\}^{\frac{1}{2}}.$$

Setting $\phi=\pi/2$ gives the radius in the plane normal to M_2:

[3.14] $$\theta = \arccos\{(m_3-c)/(m_3-m_1)\}^{\frac{1}{2}}.$$

3.4 Testing the Adequacy of the Model

3.4.1 General Statement

The vector statistics are adequate to characterize a distribution that conforms to Fisher's model, but not one of any greater complexity, for example, asymmetric or multimodal. The axis statistics are adequate to characterize a distribution that conforms to Bingham's model, but not one of any greater complexity. Thus characterization of a set of vectors by the vector statistics constitutes implicit assumption that the data distribution can be approximated by Fisher's model; similarly, characterization of a set of axes by the axis statistics constitutes implicit assumption that the data distribution can be approximated by Bingham's model.

If the distribution is more complex than the assumed model, the statistics are not adequate to completely characterize the distribution, and the data must be further subdivided. Thus it is necessary to test for distributions more complex than the assumed model. If, on the other hand, the distribution is simpler than the assumed model, the statistics are again misleading and should be simplified to represent the true form of the distribution. Thus tests for simpler cases than the assumed distribution are required.

Distributions less complicated than the assumed distribution are special cases of the assumed distribution and can be tested for by means of parametric tests based on the calculated statistics. Since more complex distributions are by definition beyond the scope of the assumed model, parametric tests are out of the question and non-parametric tests or data displays are the means that are used.

3.4.2 Vector Statistics

Tests for Distributions More Complex than Fisher's

Conditions to be looked for are multimodality and axial symmetry. Multimodal distributions must be further subdivided and the elements characterized separately. An attempt should be made to further subdivide axially symmetric distributions, particularly by areal analysis. If subdivision into axially symmetric elements is not possible, the vector statistics can be used with reservation.

Data displays are an important diagnostic tool, especially projections of the spherical distribution onto the plane normal to the computed mean direction, and rose diagrams showing the circular distribution of the projections of the vectors onto this plane. This circular distribution can be tested for uniformity either by using a chi-square test as described by Watson and Irving (1957) or using Kuiper's test described by Mardia (1972, p.178). This test has been shown to be sensitive to two-peaked and four-peaked alternatives to uniformity (Mardia, 1972, p.194). If a cluster of points on the sphere does not have axial symmetry, it is commonly because either (a) the cluster is elongated rather than circular or (b) it is actually two clusters. In either case, if the cluster is rotated until the computed mean is parallel to the z-axis, then if (θ,ϕ) are spherical coordinates referred to the axes

X, Y and Z, the resulting distribution of the ϕ_1 will have two opposite maxima. Hence Kuiper's test is suitable.

Kuiper's test is performed by computing the cumulative distribution of the data and the theoretical cumulative distribution. If l_{max} and l_{min} are respectively the maximum and minimum values of the difference between the two cumulative distributions, then Kuiper's statistic is $l=l_{max}-l_{min}$. Percentage points of the statistic

[3.15] $$n^{\frac{1}{2}}l$$

have been tabulated by Stephens (1965) and Mardia (1972, p.308). If n>8 the statistic

[3.16] $$(n^{\frac{1}{2}} + 0.155 + 0.24n^{-\frac{1}{2}})\,l$$

is independent of n; percentage points are given by Stephens (1970) and Mardia (1972, p.178).

The distribution of deviations from the computed mean can be compared with Fisher's model using a chi-square test of fit. For k>3 Watson and Irving (1957) gave the probability that a value θ will be observed that is less than θ_o as

[3.17] $$\exp\{-k(1 - \cos\theta_o)\}.$$

By dividing the range 0 to π into classes, a chi-square test of fit can be performed. The simplest way to avoid very small expected frequencies is to construct the classes so that the expected frequencies are all equal. For example, limits of 10 such classes, each having an expected frequency of n/10, can be computed by setting [3.17] successively equal to 0.9, 0.8, ... 0.1 and computing $\cos\theta_0$ in each case.

Tests for Special Cases of Fisher's Model

When k=0 Fisher's distribution reduces to the uniform distribution. The hypothesis: k=0 can be tested by referring the length m of the mean vector to tables in Watson (1956a) or Mardia (1972).

3.4.3 Axis Statistics

Tests for Distributions More Complex than Bingham's

Multimodality is the only serious condition to be looked for in this case, since axially asymmetric distributions are encompassed by Bingham's model. Data displays remain the only tool for this job. In particular, projections of the spherical distribution onto the planes normal to M_1 and M_3, and rose diagrams of the circular distributions of the projections of the axes in those planes are useful. Multimodal distributions must be further subdivided and the components characterized separately.

Tests for Special Cases of Bingham's Distribution

Important special cases of Bingham's distribution are the uniform distribution, the axially symmetric equatorial distribution, and the axially symmetric bipolar distribution. At present multi-sample tests are available only for the axially symmetric cases. The hypothesis: $k_1=k_2=k_3$ can be tested against the alternative: not all the k's are equal, by means of one of the statistics

[3.18] $$(5n/6)\sum_{i=1}^{3}\log^2(3m_i)$$

[3.19] $$(5n/6)\sum_{i=1}^{3}(3m_i-1)^2$$

both of which are distributed under the null hypothesis as chi-square with 5 degrees of freedom when n is large.

For the test for axial symmetry about the eigenvector M_3 (bipolar case) Bingham gave three statistics:

[3.20] $$n(\hat{k}_2-\hat{k}_1)(m_2-m_1)/2$$

[3.21] $$(n/3)\log^2(m_1/m_2)$$

[3.22] $$(4n/3)\{(m_2-m_1)/(m_2+m_1)\}^2.$$

All three are distributed as chi-square with 2 degrees of freedom when n is large. For the test for axial symmetry about M_1 (girdle case) Bingham gave three statistics:

[2.23] $$n(\hat{k}_3-\hat{k}_2)(m_3-m_2)/2$$

[3.24] $$(n/3)\log^2(m_2/m_3)$$

[3.25] $$(4n/3)\{(m_3-m_2)/(m_3+m_2)\}^2.$$

All three are again distributed as chi-square with 2 degrees of freedom when n is large.

There are two other tests that should be mentioned.

here. These are tests for uniformity, where now the uniform distribution is regarded as (a) a special case of the equatorial form of the Dimroth-Watson distribution and (b) a special case of the bipolar form of the Dimroth-Watson distribution. The test is performed by referring (a) m_1 and (b) m_3 to tables given by Anderson and Stephens (1972). One of these tests is appropriate when one of the cases of the Dimroth-Watson distribution can be assumed. Since they require more restrictive assumptions than Bingham's test for uniformity, they should be more powerful.

Mark (1973) used m_1 and m_3 and Anderson and Stephens' (1972) tables in testing the long axes of till stones for uniformity and for preferred orientation. In this situation axially symmetric and axially asymmetric equatorial and bipolar distributions are all possibilities. Hence one particular form of Dimroth-Watson's distribution cannot be assumed, and Anderson and Stephens' tests are not suitable. Since it is required to test for uniformity against both equatorial and bipolar alternatives and to distinguish these alternatives, Bingham's tests are quite suitable.

The rigour required in the characterization process is determined by the purpose of the characterization. While Bingham's tests are somewhat insensitive (Bingham, 1964) their failure to reject axial symmetry probably justifies acceptance of the axially symmetric model for the purpose of performing multi-sample tests. However, in some circumstances a more powerful test for axial symmetry is desirable. If the ability to distinguish a symmetric equatorial distribution from an asymmetric equatorial distribution is crucial, as it is, for example, in studies of the preferred orientations of elongated stones in glacial till or other sediments, then a test for axial symmetry is required that is powerful in distinguishing axial symmetry from the particular alternatives expected. With axis data an asymmetric equatorial distribution has at least two density maxima, and if mixed distributions are a possibility there could be four maxima. Thus Kuiper's test for uniformity on the circle, described in section 3.4.2, is appropriate in this situation. For this purpose it is applied to the circular distribution of the projections of the axes on the plane normal to M_1.

3.5 Treating Weighted Observations

We suggest the following procedures for use with weighted observations. For vector data, let the unit vectors $A_1, \ldots, A_n$ be associated with weights $w_1, \ldots, w_n$. In place of $\Sigma A_i/n$ as the mean vector it seems natural to use the weighted mean

$$[3.26] \qquad M = \Sigma w_i A_i / \Sigma w_i.$$

This is consistent with the previous definition [3.1] since when all the weights are equal it reduces to

$$M = w \Sigma A_i/nw = \Sigma A_i/n.$$

Also when all the A_i are equal to A

$$M = A\ \Sigma w_i / \Sigma w_i = A.$$

Thus 1-m can still be taken as a measure of dispersion and k can be estimated as before.

For axis data we suggest that for weighted observations the matrix T can be redefined as the weighted mean

[3.27] $$T = \Sigma w_i A_i A_i' / \Sigma w_i.$$

Again this is consistent with the previous definition [3.2] since when all the w_i are equal to w

$$T = w\ \Sigma A_i A_i'/nw = \Sigma A_i A_i'/n$$

as before. Also trace(T) = 1, and when all the A_i are equal to A

$$T = AA'\ \Sigma w_i / \Sigma w_i = AA'.$$

When observations are weighted the non-parametric tests can be performed as follows. For the chi-square test of fit, the ith observation should be counted as $nw_i/\Sigma w_i$, so that the sum of all the observed class frequencies is n. For Kuiper's test the cumulative proportion up to and including the jth observation can be calculated as the cumulative proportion of the weights, that is,

$$\sum_{i=1}^{j} w_i \Big/ \sum_{i=1}^{n} w_i.$$

3.6 Computer Programs

The program MODEL1 computes the vector and axis statistics and performs the parametric tests for special cases based on the axis statistics. If axial symmetry is accepted, the parameter of the Dimroth-Watson model is estimated. In all cases the data are subjected to two rotations. The first rotation brings the eigenvectors M_1, M_2 and M_3 into coincidence with the reference axes north, east and down respectively. In this orientation the direction of maximum density is vertical. The second rotation brings the eigenvectors M_3, M_2 and M_1 into coincidence with the reference axes north, east and down respectively. In this orientation the direction of minimum density is vertical. For each orientation, the program computes the new trends of the data points and Kuiper's test statistic. Printed output consists of the various statistics and test results. Other output consists of the data in its original form, the data in the two rotated forms, and the two circular frequency distributions corresponding to the two new orientations.

Using this output, the program WNDPLOT3 can produce equal-area projections of the spherical distribution of point density, for the data in its original orientation and the two new orientations. This program operates as follows. To map the density of points on the sphere, a density estimate is computed at each of a number of predetermined "counting locations". For unweighted points, the density estimate at a particular location would be obtained by counting the number of points falling inside a predetermined area (the "counting circle") surrounding this location. The density estimate is usually obtained by expressing this count as a percentage of the total number of points. This program obtains a density estimate for weighted points by summing the weights of the points falling inside the area and expressing this sum as a percentage of the total of the weights of all the data points. The density estimates are printed in the form of an equal-area projection.

The program uses a print matrix consisting of 3713 print positions: 47 lines with 79 characters on each line. Printed six lines to the inch, and ten characters to the inch horizontally, the print matrix is approximately eight inches square. The equal-area projection is a circle with a 10-cm radius centred on the print matrix. Every print position whose centre lies within the circle represents a counting location. There are 2933 counting locations. The counting locations are defined by direction cosines, which are read in by the program at the start of each run.

With such a large number of counting locations it is inefficient to compare every counting location with every data point in the counting process. The number of comparisons is reduced as follows. For a particular data point, those counting locations that are to be incremented lie within a circle on the sphere whose projection approximates an ellipse that has its greatest size when the data point lies in the horizontal plane. Before beginning the counting procedure, the program computes the maximum dimension, c, for the ellipse in this case. For each data point, the program computes x and y, the coordinates on the projection of the projected data point, then determines the submatrix of the print matrix that lies inside the square bounded by x+c, x-c, y+c, y-c. This square must wholly contain the projection of the counting circle centred on the data point. Each counting location represented in this submatrix is then tested by computing its angular distance from the data point and comparing this with the radius of the counting circle. This process is repeated in a slightly modified form if the data point is close enough to the horizontal to affect counting locations on the diametrically opposite part of the projection. In this way a great many comparisons are eliminated by means of a relatively simple computation.

After the counting procedure is completed, the program constructs the plot, line by line, by expressing each count as a percentage of the total weight and then placing the appropriate character in the corresponding print position. A contour interval and a string of characters are supplied to the program on parameter cards. As an enhancement to the visual impact of the plot, characters representing higher densities are overprinted, the number of overprintings increasing with the density value.

The program NFPLOT1 can use the same output from MODEL1 to produce the corresponding equal-area projections of the spherical point distributions. In this program the coordinates of points on the projection are computed from direction cosines using formulas derived as follows. Let a direction have spherical coordinates (θ,ϕ) (trend ϕ, plunge $(\pi/2-\theta)$) and direction cosines x, y and z. On a Lambert equal-area projection with radius u_o let the projection of the point (θ,ϕ) lie at a distance u from the centre of the projection and let x_p and y_p be the coordinates of the projected point referred to the centre of the projection, where x_p and y_p are measured parallel to the X and Y axes respectively. Then, from the definition of the equal-area projection,

$$u = 2^{\frac{1}{2}} u_o \sin(\theta/2)$$

or

[3.28] $$u = u_o(1 - \cos\theta)^{\frac{1}{2}}.$$

and

[3.29] $$x_p = u \sin\phi$$

[3.30] $$y_p = u \cos\phi.$$

Substituting [3.28] in [3.29] and [3.30]

[3.31] $$x_p = \sin\phi \, u_o(1 - \cos\theta)^{\frac{1}{2}}$$

[3.32] $$y_p = \cos\phi \, u_o(1 - \cos\theta)^{\frac{1}{2}}.$$

From [2.26], [2.27] and [2.28] we have

[3.33] $$\cos\theta = z$$

[3.34] $$\sin\theta = (1-z^2)^{\frac{1}{2}}$$

[3.35] $$\cos\phi = x/(1-z^2)^{\frac{1}{2}}$$

[3.36] $$\sin\phi = y/(1-z^2)^{\frac{1}{2}}.$$

Now combining [3.31] with [3.33] and [3.36]

$$x_p = \{y/(1-z^2)^{\frac{1}{2}}\}u_o(1-z)^{\frac{1}{2}}$$

[3.37] $$= yu_o\{(1-z)/(1-z^2)\}^{\frac{1}{2}}.$$

Combining [3.32] with [3.33] and [3.35]

$$y_p = \{x/(1-z^2)^{\frac{1}{2}}\}u_o(1-z)^{\frac{1}{2}}$$

[3.38] $$= xu_o\{(1-z)/(1-z^2)\}^{\frac{1}{2}}.$$

Finally, the program ROSE uses the circular frequency distributions written by MODEL1 to produce smoothed rose diagrams as follows. Input to the program ROSE is a vector of relative frequencies in 360 1-degree intervals. Let t_i, $i=1,\ldots,360$, be the observed relative frequencies. Then the smoothed relative frequency for any given interval is

[3.39] $$\sum_{i=1}^{360} t_i w(\theta_i) / \sum_{i=1}^{360} w(\theta_i)$$

where w is a weighting function and θ_i is the angle between the given interval and the i^{th} interval. w is taken to be the von Mises density function which is proportional to

$$\exp(k\cos\theta)$$ (Mardia, 1972, p.57).

This is analogous to the Fisher-weighted density estimator described in Chapter 2.

All the problems related to the amount of smoothing to be applied to the data that were discussed in section 2.6.3 arise again in connection with smoothing of rose diagrams. The statements made in 2.6.3 about optimization of the amount of smoothing apply equally here. Again, the size of the errors in the data probably should play some part in determining the amount of smoothing to be used. In the examples given here, the von Mises density function with k=50 was used as the smoothing or weighting function for no other reason than that it produced rose diagrams that "looked reasonable". Future research in this area can be expected to provide a meaningful basis for choosing the smoothing function taking into consideration such factors as sample size and precision of measurements.

Computer programs are documented in Appendix 1.

3.7 Numerical Examples

Output from the program MODEL1 for 297 weighted poles to bedding from Domain 3 at the Fording coal mine is shown in Table 1. Figure 25 shows a lower-hemisphere equal-area point diagram of the poles. Bingham's tests at the 5% level reject uniformity but accept axial symmetry about M_3 (bipolar case). The parameter of the best-fitting bipolar Dimroth-Watson distribution is 35.4. The modified Kuiper's test rejects uniformity in the plane normal to M_3 at the 5% level.

The method of characterizing distributions of unknown form is illustrated with three samples of long axis orientations of elongated till stones. The data were kindly supplied by Dr J A Westgate, and were obtained from a hummocky disintegration moraine near Cooking Lake about 15 miles east of Edmonton (Bayrock and Hughes, 1962). It is of interest to know whether the axes are randomly oriented or whether they tend to lie in a plane, cluster about a line, or both. The eigenvalues of T for each sample are given in Table 2. A lower-hemisphere projection of the axes in sample 30 is shown in Figure 26A. Figure 26B shows the same data with the plane of projection normal to M_3, while in Figure 26D the plane of projection is normal to M_1. Figures 26C and 26E show the smoothed rose diagrams corresponding to the projection planes of Figures 26B and 26D respectively. For this sample the statistics for testing for uniformity [3.18] and [3.19] are respectively 63.9 and 44.7. The upper 5% point of the chi-square distribution with 5 degrees of freedom is 11.07. Thus the hypothesis of uniformity is strongly rejected. To test the hypothesis of axial symmetry about M_1 we use [3.24] and [3.25] that do not require estimation of k_1 and k_2; these statistics are respectively 4.82 and 4.69. The upper 5% point of the chi-square distribution with 2 degrees of freedom is 5.99; thus the hypothesis of a uniform equatorial distribution is not rejected. For the test of axial symmetry about M_3 the statistics [3.21] and [3.22] have values respectively of 21.37 and 19.07. Thus axial symmetry about M_3 is rejected. Hence for sample 30 Bingham's tests lead to acceptance of the uniform equatorial model. However, after projection onto the plane normal to M_1 the modified Kuiper's statistic is 1.85. Since the upper 5% point of this statistic is 1.747, we are led to reject the uniform equatorial model after all. Further analysis of the distribution in this plane would identify the preferred directions (Figures 26D and 26E). If a unimodal distribution is inferred from this analysis, the distribution can be characterized by the three-dimensional axis statistics already obtained. If a multi-modal distribution is inferred, it should be subdivided if possible and the analysis repeated for each fabric element.

Point diagrams and rose diagrams for sample 100 are shown in Figure 27 . For this sample the statistics [3.18] and [3.19] have values 39.21 and 34.06, so the hypothesis of uniformity is rejected. The statistics for axial symmetry about M_1 are 8.24 and 7.60, while those for axial symmetry about M_3 are 6.43 and 6.03. Thus all three hypotheses are rejected for this sample. This sample illustrates the danger of relying entirely on numerical testing procedures. Kuiper's test accepts axial symmetry about M_3, whereas the rose diagram (Figure 27C) clearly shows asymmetry about this axis. Kuiper's test rejects axial symmetry about M_1 (Figure 27D and E). This sample thus needs further subdivision before it can be characterized.

For sample 110 the displays are shown in Figure 28. For this sample the uniformity statistics are 3.96 and 4.29, so that the hypothesis of uniformity is accepted.

3.8 Summary and Conclusions

If the data are vectors, they should be characterized by the vector statistics. The adequacy of these statistics can be checked by means of data displays and both parametric and non-parametric tests. If this characterization is adequate, multi-sample tests can be used. If not, the data must be further subdivided.

If the data are axes, they should be characterized by the axis statistics. The adequacy of this characterization should be checked by means of data displays and parametric tests for special cases. The data displays and parametric tests recommended here provide a versatile means for characterizing a set of axes thought to constitute a single fabric element. Multi-sample tests are available for axially symmetric cases only.

Fisher's model can be used to obtain meaningful results with axis data by assigning sense or direction to the axes provided that

(1) the axes have a unimodal axially symmetric distribution, as indicated by an adequate fit to Fisher's model,

(2) the dispersion of the axes is sufficiently small (estimated concentration parameter of Fisher's model >3, which means that $\Pr[\theta > \pi/2] \approx 0.05$), and

(3) sense is assigned to the axes so that the resulting vectors form a single cluster.

The smoothed rose diagram produced by the program ROSE is a new display technique, and Bingham's single-sample tests are introduced for the first time to fabric analysis.

The adequacy of a characterization depends in part on the purpose of the characterization. For example, Bingham's tests are suitable for establishing axial symmetry for the purpose of performing the multi-sample tests described in Chapter 4. However, in some studies the importance of identifying preferred orientations in an equatorial distribution calls for a more powerful test. Kuiper's non-parametric test used to test for uniformity on the circle is probably the most suitable. Similar remarks apply to Bingham's test for uniformity.

Weighted observations, whether axes or vectors, can be processed by means of the modified procedures detailed in section 3.5. These procedures are new.

The tests based on Bingham's distribution, including the multi-sample tests discussed in Chapter 4, are introduced here for the first time to fabric analysis. The introduction of the multi-sample tests for axially symmetric cases makes available for the first time a set of testing procedures for axis data corresponding to those available for vector data. This is an important advance in fabric analysis, since most fabric data are axes. The axis statistics, hitherto little used, are greatly increased in importance and usefulness with the availability of testing procedures based on them.

4 Areal Analysis

4.1 Introduction

Characterization of the fabric of a body of rock requires the definition of domains within which the fabric of the body is homogeneous. In this chapter this problem is considered in relation to a single fabric element. In practice this analysis would be carried out for each measured fabric element. In the statistical programs the tests based on Fisher's distribution for vectors with a polar distribution have been used. The tests for axis data with either axially symmetric bipolar or axially symmetric equatorial distributions are also given in the text.

The usual procedure for areal analysis of fabric data was described in Chapter 1. A method based on separate plots of dip direction and dip has been described by Piteau and Russel (1971). The cumulative deviations of observed dip directions from the mean dip direction were plotted against distance along the exposed face. A similar plot was produced for dips. From these plots the locations at which the fabric changed could be determined, and structural regions or domains thus defined. The application of this technique to problems of characterizing and extrapolating joint sets for engineering purposes has been discussed by Piteau (1973). Another technique, proposed by Elliot (1965), involves the use of "isogonic" maps. These are maps showing contours of equal angles or isogons, the angle that is contoured being chosen according to the nature of the data, for example, dip of surfaces or pitch of lineations. Regions of homogeneous fabric are identified on the isogonic map as regions of low density of isogons, whereas high isogon density indicates highly variable fabric.

Areal variation of two-dimensional orientation data has been studied by applying normal and von Mises distribution theory (Agterberg and Briggs, 1963). Two-dimensional vector trend analysis has been used by Agterberg, Hills and Trettin (1967) in the analysis of paleocurrent indicators, and by Roberts and Mark (1970) in the analysis of till fabrics. A computer program for trend analysis of two dimensional vector data has been given by Fox (1967). The moving average approach to the analysis of paleocurrent indicators was taken by Sturm (1971).

In this chapter we describe a computer-based procedure for analyzing areal variation of three dimensional fabric data that offers the following advantages.

(1) The area is divided by the geologist into sampling units. Selective retrieval of the data and all data manipulation are performed by the computer.

(2) The data are treated throughout in a three-dimensional framework, so that it is not necessary to treat horizontal and vertical angles in separate analyses.

(3) A statistical test is provided that indicates those sampling units within which the scatter in the orientations of the measured structures making up the fabric element is significantly large, thus providing a means for identifying locally disturbed fabrics.

(4) Subfabrics of the sampling units are compared by means of a statistical test for homogeneity, so that decisions are made on a quantitative basis, and results are objective and reproducible. A computer program HOMO has been written to implement both of these tests.

(5) Fabric diagrams for the sampling units are produced by the computer; a computer program WNDPLOT3 has been written to produce these diagrams on the line printer.

(6) Since all the laborious manual operations including data retrieval are performed by the computer, combination, testing and re-combination of the sampling units can be performed rapidly even when many observations are involved.

In addition to these advantages, a new display technique has been devised. This consists of a map on which are displayed the mean orientations for the sampling units.# A given orientation is displayed on the map as a line whose direction and length are those of a line drawn on an equal-area projection from the centre of the projection to the point representing the given orientation. Either the mean orientations for the sampling units or the individual observations can be displayed. The display can be made to show either true orientations or deviations from an overall mean. These deviations may be thought of as the new orientations after rotation to make the overall mean vertical. A program MAP has been written to produce the display on the Calcomp drum plotter. So that the display can be used in conjunction with an equal-area point diagram, the program NPPLOT1 has been written to produce equal-area point diagrams on the Calcomp plotter.

First, multisample tests based on the models described in Chapter 3 are presented in section 4.2. These tests, except those based on Fisher's model, are introduced here for the first time to fabric analysis. In the case of the Dimroth-Watson distribution, we have generalized the two-sample tests given by Watson (1965) and Mardia (1972) to the multisample case. Following this, in section 4.3, the computer-based procedure is described in detail, including the new display technique. Section 4.4 presents a worked example, and section 4.5 summarizes the chapter.

4.2 Multisample Tests

4.2.1 Fisher's Model

The following multisample tests based on the Fisher distribution were given by Mardia (1972). Let A_{ij}, $i=1,\ldots,n_j$, $j=1,\ldots,q$, be q independent random samples of sizes n_j drawn from Fisher distributions with concentration parameters k_j. Let $n=\Sigma n_j$, m_j be the length of the mean vector of the j^{th} sample and m the length of the mean vector of the combined sample.

A test for the homogeneity of concentration parameters may be performed as follows.

Case 1. $m\leq 0.44$. Under the null hypothesis

$$[4.1] \qquad \Sigma g_j h_j^2 - (\Sigma g_j h_j)^2/\Sigma g_j \simeq \chi^2_{q-1}$$

where

$$g_j = (5/3)(n_j-5) \text{ and } h_j = \arcsin(3m_j/5^{\frac{1}{2}}).$$

Case 2. $0.44<m<0.67$. The same statistic as [4.1] where now $g_j^{-1}=0.39356/(n_j-4)$ and $h_j=\arcsin\{(m_j+0.17595)/1.02903\}$.

Case 3. $m\geq 0.67$. Under the null hypothesis

$$[4.2] \qquad [g \ln\{(n-\Sigma n_j m_j/g\}-\Sigma g_j \ln\{n_j(1-m_j)/g_j\}]/(1+h) \simeq \chi^2_{q-1}$$

where now

$$g_j=2(n_j-1),\ g=2(n-q) \text{ and } h=(\Sigma g_j^{-1}-g^{-1})/\{3(q-1)\}.$$

Assuming that $k_1=k_2=\ldots=k_q=k$ (k unknown), a test for the equality of means may be performed as follows.

Case 1. $m\geq 0.32$. In this case

$$[4.3] \qquad (1-1/5\hat{k}^2)(n-q)(g-m)/\{(q-1)(1-g)\} \simeq F_{2q-2,2n-2q}$$

where $g = \Sigma n_j m_j / n$ and $\hat{k}$ is given by

$$\coth(\hat{k}) - 1/\hat{k} = m$$

which is solved using the table in Mardia (1972, p.322) and which, for $m > 0.9$, becomes

$$\hat{k} = (1-m)^{-1}.$$

[4.3] is the approximation given by Watson and Williams (1956) with the addition of the correction factor for small k. If $m \geq 0.67$, the term in $\hat{k}$ can be omitted.

Case 2. $m < 0.32$. In this case

[4.4] $$(1-\hat{k}^2/15+3q/2n\hat{k}^2)^{-1}(3n)(g^2-m^2) \simeq \chi^2_{2q-2}$$

where again $g = \Sigma n_j m_j / n$ and $\hat{k}$ is as before. If n is small, this approximation is satisfactory if $\hat{k}$ is not near 0 or 1; if n is large the term in $\hat{k}$ can be omitted.

While Mardia (1972, p.270) gave the null distribution of [4.2] as chi-square with 2q-2 degrees of freedom, we have used chi-square with q-1 degrees of freedom for the following reason. Mardia's test is based on Bartlett's test for equality of variances of samples from normal distributions (Mardia, 1972, p.166). Now Bartlett's test statistic is an expression for $2\ln\lambda$ involving estimates of population parameters where λ is the likelihood ratio for the alternative and hypothesis models. The test depends on the fact that as n increases without limit the distribution of $2\ln\lambda$ approaches the chi-square distribution with df degrees of freedom, where df is the difference between the number of parameters in the alternative model and the number of parameters in the hypothesis model (Fraser, 1958, p.255-260). In this case the alternative model consists of q samples with different means and different dispersions, and the hypothesis model consists of q samples with different means but the same dispersion, so the difference between the numbers of parameters is q-1, and $2\ln\lambda$ is asymptotically distributed as chi-square with q-1 degrees of freedom. In order to verify this, 20 samples of 10 points each were generated from a Fisher distribution with k=50 (using the program SAMPLEGEN documented in Appendix 1), and the statistics [4.2] and [4.3] were computed using the program HOMO1. This procedure was performed 10 times. The ten values of the two statistics are given in Tables 3 and 4, with the results of tests of fit to the null distribution given by Mardia. The statistic [4.3] for homogeneity of means fitted the F distribution with 2q-2=38 and 2n-2q=360 degrees of freedom (Table 3). However, the statistic [4.2] for homogeneity of concentration parameters was consistently much too small to fit the chi-square distribution with 2q-2=38 degrees of freedom. The values of the statistic [4.2] were compared with the chi-square distribution with q-1=19 degrees of freedom; the values fit this distribution (Table 5).

The above multisample tests based on Fisher's model have been implemented in the program HOMO1 documented in Appendix 1.

4.2.2 Dimroth-Watson's Model

Bipolar Case

Mardia (1972) has given the result

$$2k(n-m_3) \simeq \chi^2_{n-2}$$

for the bipolar case of the Dimroth-Watson model, where k is the concentration parameter, n is the sample size and m_3 is the largest eigenvalue of the matrix T defined in Chapter 3. We use this result to derive a test for the equality of concentration parameters of two samples. For two samples of sizes n_1 and n_2 from populations with concentration parameters k_1 and k_2 we have

$$2k_1(n_1-m_{31}) \simeq \chi^2_{n_1-2}$$

$$2k_2(n_2-m_{32}) \simeq \chi^2_{n_2-2}$$

where m_{31} and m_{32} are the largest eigenvalues of T for samples 1 and 2. If $k_1=k_2$ we obtain by division

$$[4.5] \qquad (n_2-2)(n_1-m_{31})/(n_1-2)(n_2-m_{32}) \simeq F_{n_1-2,n_2-2}$$

which can be used to test the equality of k_1 and k_2. For more than two samples, the maximum F-ratio (Hartley, 1950) can be used. If q samples yield estimates $\hat{k}_1,\ldots,\hat{k}_q$, the ratio of the largest to the smallest $\hat{k}_j$ is referred to tables in David (1952) using the average of the degrees of freedom for the two samples (Hartley, 1950).

To test the equality of the means of two samples assuming equality of concentration parameters, Mardia (1972) has given the statistic

[4.6] $(n-4)(m_{31}+m_{32}-m_3)/2(n-m_{31}-m_{32}) \simeq F_{2,n-4}$

where m_{31}, m_{32} and m_3 are respectively the largest eigenvalues of T for sample 1, sample 2 and the combined sample, and $n=n_1+n_2$. Continuing to follow the analysis of variance analogy (Watson, 1965), we generalize [4.6] to the case of q samples to obtain

[4.7] $(n-2q)(\Sigma m_{3j}-m_3)/(2q-2)(n-\Sigma m_{3j}) \simeq F_{2q-2,n-2q}$

where Σ means the sum for $j=1,\ldots,q$.

Girdle Case

Watson (1965) gave the result

$$2km_1 \simeq \chi^2_{n-2}$$

For the girdle case of the Dimroth-Watson distribution with parameter k, where m_1 is the smallest eigenvalue of T. As before, we use this result to obtain the test for equality of concentration parameters of two populations:

[4.8] $(n_2-2)m_{11}/(n_1-2)m_{12} \simeq F_{n_1-2,n_2-2}$

where m_{11} and m_{12} are the smallest eigenvalues for samples 1 and 2 respectively. Again, the maximum F-ratio can be used when there are more than two samples.

The test for equality of means of two populations was given by Watson (1965) as

[4.9] $(n-4)(m_1-m_{11}-m_{12})/2(m_{11}+m_{12}) \simeq F_{2,n-4}$

where m_1 is the smallest eigenvalue of T for the combined sample and $n=n_1+n_2$. Again we extend this result to the case of q samples to obtain

[4.10] $(n-2q)(m_1 - \Sigma m_{1j})/(2q-2)(\Sigma m_{1j}) \simeq F_{2q-2,n-2q}$

where m_{1j} is the smallest eigenvalue of T for the j^{th} sample and Σ means the sum for $j=1,\ldots,q$.

4.3 Computer-Based Procedure

The procedure is illustrated by the flow diagrams in Figures 29 to 35. The data representing the isolated fabric element whose areal variation is to be studied are placed in a separate data file by retrieval from the master data file using program KEY4. The program DATFIX1 ensures that every observation is associated with a set of coordinates and that the observations on each traverse appear in the file in order of increasing distance from the traverse origin. The program TRVLST1 then generates a list of the traverses that is used by several other programs. The first program to use the information in this traverse file is the program MAP3, which produces a map of the traverses on the plotter (Figure 29). This map includes the number of observations on each traverse. Note that only those traverses are shown on which observations were made that appear in the retrieved data file currently undergoing areal analysis; also that the number of observations shown for each traverse is the number of observations made on the traverse that appear in that data file.

At this stage the data file is processed by the characterization program MODEL1 described in Chapter 3, so that a complete statistical summary of the data is available (Figure 30B). If the data are found to fit one of the model distributions the areal analysis procedure is still worth while, since it will reveal areas of high variability or large deviations from the mean. At this point various displays can be produced of all the data, including equal-area projections both of points and of point density, and a map showing locations and orientations of individual observations (Figures 32 and 34). On this map the geologist outlines the sampling units that are to serve as the basic units in the analysis. The analysis can be re-started at this point in case redefinition of sampling units is found necessary.

Once the sampling units have been outlined, the information is made available to the computer in the form of traverse numbers and observation numbers on each traverse. The program GRPKEY1 then produces a "key" for each sampling unit, that is, a list of the numbers of the lines in the data file that contain the observations in each sampling unit (Figure 30). The sampling units are immediately processed one at a time by the program GRPSTA1, that computes a mean direction and a concentration parameter for each sampling unit. These results are formed into the sample data file (called the group data file in Figure 31) and form the basis for the statistical tests for homogeneity. At this point displays of the sample means are produced, both equal-area projections and a map showing locations and orientations of the sample means (Figures 32 and 34). This map also shows the concentration parameters for the samples.

HOMO1 is an interactive program that tests for homogeneity of both mean directions and concentration parameters (Figure 31). It does this initially for all the samples. Since homogeneity of concentration parameters must be established before tests for mean directions are valid, the first task of the geologist carrying out the analysis is to instruct the program HCMO1 to delete from the analysis those samples that have large dispersions. After each deletion the remaining concentration parameters are tested for homogeneity. When homogeneity of concentration parameters is achieved, a list of deleted samples is obtained so that these can be flagged for further field investigation, since these are the locations at which the fabric is exceptionally variable.

At this point a new map of the remaining sample means can be produced that shows the deviations of the sample means from the overall mean (Figures 33 and 34). A similar display of the individual observations can also be produced. Any tendency for the orientations to deviate from the overall mean in a consistent way in one part of the area should be apparent in this display. Tentative homogeneous regions can now be outlined on this map, and tested using the program HOMO1 by giving the program the numbers of the sampling units to be grouped (these numbers are shown on the map of sample means). Sampling units can be added to or deleted from the tentative grouping one at a time and homogeneity retested until the largest possible homogeneous regions have been established. Finally, a complete characterization is produced for each homogeneous region by the program MODEL1 (Figure 35).

4.4 Worked Example

Analysis of Fording Domain 3 Bedding

Initial statistics on this data indicate good clustering (Fisher concentration parameter 68), but Kuiper's test indicates lack of axial symmetry (statistic = 1.845, 95% point = 1.747). This asymmetry is however not marked enough to be detected by Bingham's tests based on the eigenvalues (statistic = 3.0, 95% point = 6.0).

The poles to the observed bedding surfaces are displayed on an equal-area projection in Chapter 3 (Figure 25). A second projection (Figure 36) shows the poles after rotation to bring the mean pole (trending 253 degrees, plunging 73 degrees) into a vertical position. (Note that the axis of rotation is not horizontal. The rotation is such that the eigenvectors, in order of increasing eigenvalues, are brought into coincidence with the reference directions north, east and down.)

These poles are also displayed on a map (Figure 37). Each point on the map indicates the location of an observed bedding surface;the line drawn from the point has the same azimuth and length as a line drawn on Figure 25 from the centre of the projection to the projected pole.

The traverses on which the data were collected are shown on the traverse map (Figure 38). The observations were grouped into 32 samples, with the number of observations in the samples ranging from 6 to 12; most samples contained from 8 to 10 points. Fisher concentration parameters for the samples range from 14 to 2112 (Table 6). The rotated sample means are shown in Figure 39. The test for homogeneity of concentration parameters initially gave $\chi^2 = 314.97$, df=62. The probability of obtaining a statistic equal to or greater than this value under the null hypothesis is less than 0.005. The test of sample means initially gave F = 12.09, df=42,368, probability less than 0.005. This clearly demonstrates that the part of the Fording mine designated "Domain 3" is not a domain with respect to the bedding subfabric.

4.5 Summary and Conclusions

The computer-based procedure for analyzing the areal variation of an isolated fabric element described here offers substantial advantages over conventional procedures. Definition of domains is based on statistical tests for homogeneity of mean directions and dispersions of local samples. This method has the advantage that locations of exceptionally high fabric variability or exceptionally large deviations from the mean orientation are identified. Definition of domains is assisted by a new display that simultaneously shows the location and three-dimensional orientation of either an individual observation or a local mean direction. The display can be made to show deviations from an overall mean direction, so that regions of consistent deviation from the overall mean can be readily identified as distinct fabric domains. An interactive program and completely computerized data retrieval and handling system permit testing of tentative domains to be carried out very rapidly, especially if a time-sharing computer system is available. This means that several hypotheses about the data can be investigated statistically without large investments of time for data retrieval and plotting of diagrams.

Multisample tests for axis data based on the axis statistics are introduced here for the first time to fabric analysis. Both bipolar and equatorial distributions of axes can now be tested.

5 Summary

5.1 General Statement

This thesis describes an approach to fabric analysis that differs from the conventional approach in making much more extensive use of numerical methods in all stages of the procedure. The techniques recommended, many of them new or applied for the first time in fabric analysis, are summarized below.

5.2 Data Collection, Storage and Retrieval

Since computer processing is a prerequisite to the use of numerical methods with other than very small data files, data must be stored in machine-readable form and should be collected on field data sheets designed with that end in view. It is extremely important that analytical requirements be anticipated in the design of the data collection scheme. Data storage and retrieval programs suitable for use with fabric data are provided.

5.3 Data Display - Contoured Orientation Diagrams

The equal-area projection showing contours of point density is the most important display tool for fabric data. An attempt was made to gain a more complete understanding of this display.

Conventional techniques of calculating the density of sample points at a location on the unit sphere represent a special case of the class of statistical density estimators known as the "Parzen" class of density estimators. The special case is that in which the weighting function is a rectangular function having a constant value for distances less than a specified distance and zero elsewhere. The specified distance corresponds to the radius of the counting circle in methods conventionally used by geologists. This estimator is here referred to as the "constant-area" estimator.

The existing theory of density estimators suggests two principal avenues of exploration: (1) the use of Parzen estimators with other than rectangular weighting functions, and (2) the use of estimators belonging to the other principal class so far considered in the mathematical literature, that is, the "k-nearest-neighbour" estimators. Accordingly, two new methods of estimating population density on the sphere have been investigated in the present work: one is a k-nearest-neighbour estimator referred to here as the "constant-proportion" estimator, the other a

Parzen estimator that uses as a weighting function Fisher's probability density function, referred to here as the "Fisher-weighted" estimator. This function was chosen because it leads to an estimate algebraically identical to that obtained by assuming data points to have Fisher-distributed errors; this has been shown to be a reasonable assumption.

Theoretical investigation of these estimators (including the conventional estimator) applied to unimodal Fisher-distributed populations leads to exact equations relating the peak density of the population (and hence the concentration parameter) to the mean estimated peak density.

Experimental investigation of the estimators at off-peak as well as peak locations - again using only unimodal Fisher-distributed populations - reveals basic differences between the two classes of estimators in the way in which the properties of the estimator vary with distance from the population peak. In particular, the coefficient of variation of the Parzen estimators tends to change from very small to very large with increasing distance from the population peak. The coefficient of variation of the constant-proportion estimator, on the other hand, tends to be much less extreme, and in fact decreases slightly with increasing distance from the peak.

The two principal functions of the orientation diagram in fabric analysis are, first, to suggest a type of model for the population, and, second, to provide estimates of model parameters. We have here considered the problem of estimating the concentration parameter of a Fisher-distributed point cluster in a multimodal population. The density function of the population consists of a sum of scaled Fisher distributions, and is subjected to smoothing and random variation to produce the empirical density function which is seen on the density diagram.

The exact relationships obtained in the present study permit the concentration parameter to be estimated from the smoothed (unscaled) peak density of a Fisher-distributed point cluster for any estimator of the types investigated. The problem of scaling can be solved either by integrating on the density diagram (graphically or by simple numerical summation) or by using a measure of shape that is related to the smoothed peak density but is independent of scaling.

5.4 Numerical Characterization

Isolated fabric elements should be characterized by statistics appropriate to the data type: techniques for characterizing axially symmetric clusters of vectors are already in use in fabric analysis; techniques for characterizing axially symmetric clusters of axes, axially symmetric girdles of axes, and axially asymmetric clusters or girdles of axes have not been used in fabric analysis but are available and are described here. For such statistics to be meaningful the adequacy of the characterization must be tested using combinations of data displays and statistical tests, both parametric and non-parametric. Multi-sample tests are available for all the axially symmetric cases - clusters of vectors, clusters and girdles of axes. Most of the tests described for axis data are new to fabric analysis. A new technique is described for producing smoothed rose diagrams as an aid to checking the adequacy of characterization. New procedures are described for characterizing and testing weighted observations.

5.5 Areal Analysis

The computer-based procedure for analyzing the areal variation of an isolated fabric element described here offers substantial advantages over conventional procedures. Definition of subfabric domains is based on statistical tests for homogeneity of mean directions and dispersions of local samples. This method has the advantage that locations of exceptionally high fabric variability or exceptionally large deviations from the mean orientation are identified. Definition of domains is assisted by a new display that simultaneously shows the location and three-dimensional orientation of either an individual observation or a local mean direction. The display can be made to show deviations from an overall mean direction, so that regions of consistent deviation from the overall mean can be readily identified as distinct fabric domains. An interactive program and completely computerized data retrieval and handling system permit testing of tentative domains to be carried out very rapidly, especially if a time-sharing computer system is available. This means that several hypotheses about the data can be investigated statistically without large investments of time for data retrieval and plotting of diagrams.

```
FORDING DOMAIN 3 BEDDING
  297 POINTS WITH TOTAL WEIGHT OF 1568.1

"BINGHAM" STATISTICS

 EIGENVAL   EVAL/SW     DIRECTION   COSINES          TR   PL      DD   DP

    20.23  0.01290 -0.56992  0.80149  0.18115  125.4 10.4  305.4 79.6

    24.10  0.01537  0.81764  0.53125  0.22190   23.0 12.8  213.0 77.2

  1523.75  0.97173 -0.08161 -0.27457  0.95810  253.4 73.4   73.4 16.6

TEST OF HYPOTHESIS K1=K2=K3: TEST STATISTIC= 5243.8984     5% POINT=11.0705

TEST OF HYPOTHESIS K2=K3: TEST STATISTIC= 1702.3879     5% POINT=5.99147

TEST OF HYPOTHESIS K1=K2: TEST STATISTIC=    3.0383     5% POINT=5.99147

HYPOTHESIS K1=K2 ACCEPTED

    K FOR           CONFIDENCE RADII
    CLUSTER           95%        99%
    35.377          8.4  8.4   10.5 10.5

TEST OF UNIFORMITY ABOUT EIGENVECTOR 3

KUIPER'S STATISTIC =      1.845      95% POINT = 1.747

TEST OF UNIFORMITY ABOUT EIGENVECTOR 1

KUIPER'S STATISTIC =     12.860      95% POINT = 1.747


FORDING DOMAIN 3 BEDDING                             RO
  297 POINTS WITH TOTAL WEIGHT OF 1568.1

"FISHER" STATISTICS

     R        R/SW       DIRECTION    COSINES         TR    PL       DD    DP
  1545.26  0.98546 -0.08043 -0.27388  0.95839  253.6 73.4      73.6 16.6

   FISHER     CONFIDENCE RADII
     K           95%     99%
   68.327        1.0     1.2

TEST OF UNIFORMITY ABOUT FISHER MEAN

KUIPER'S STATISTIC =      2.559      95% POINT = 1.747
```

Table 1. Output from Program MODEL1 for Fording Domain 3 Bedding

sample	n	m_1	m_2	m_3
030	90	0.1471	0.3421	0.5108
100	49	0.1495	0.2803	0.5702
110	50	0.2792	0.3009	0.4198

Table 2. Sample sizes and eigenvalues for the till samples.

value	cumulative observed	proportions theoretical	difference
0.71	0.1	0.102	0.002
0.83	0.2	0.245	0.045
0.90	0.3	0.354	0.054
0.93	0.4	0.405	0.005
0.96	0.5	0.467	0.033
1.06	0.6	0.625	0.025
1.09	0.7	0.662	0.038
1.14	0.8	0.727	0.073
1.14	0.9	0.732	0.168
1.27	1.0	0.863	0.137

Table 3. Kolmogorov test of fit of 10 values of the statistic [4.3] to the F-distribution with 38 and 360 degrees of freedom. Maximum deviation = 0.168. Critical value for a 5% test = 0.410. Null hypothesis accepted.

	cumulative	proportions	
value	observed	theoretical	difference
06.14	0.1	0.000	0.100
10.42	0.2	0.000	0.200
13.54	0.3	0.000	0.300
16.84	0.4	0.001	0.399
16.95	0.5	0.001	0.499
17.49	0.6	0.002	0.598
24.62	0.7	0.046	0.654
25.46	0.8	0.060	0.740
28.76	0.9	0.140	0.760
30.38	1.0	0.194	0.806

Table 4. Kolmogorov test of fit of 10 values of the statistic [4.2] to the chi-square distribution with 38 degrees of freedom. Maximum difference = 0.806. Critical value for a 5% test = 0.410. Null hypothesis rejected.

	cumulative proportions		
value	observed	theoretical	difference
06.14	0.1	0.00	0.10
10.42	0.2	0.06	0.14
13.54	0.3	0.19	0.11
16.84.	0.4	0.40	0.00
16.95	0.5	0.41	0.09
17.49	0.6	0.44	0.16
24.62	0.7	0.83	0.13
25.46	0.8	0.85	0.05
28.76	0.9	0.96	0.03
30.38	1.0	0.95	0.05

Table 5. Kolmogorov test of fit of 10 values of the statistic [4.2] to the chi-square distribution with 19 degrees of freedom. Maximum difference = 0.16. Critical value for a 5% test = 0.410. Null hypothesis accepted.

group number	number of observations	concentration parameter	traverse numbers
01	08	068	159,73,74,161
02	08	597	75
03	10	056	162,163
04	07	157	64,149,150
05	09	016	65,151
06	10	228	66,152,67
07	11	524	153
08	15	124	68
09	08	779	154
10	06	226	69
11	08	114	155
12	08	227	70
13	12	106	136,54
14	07	019	137
15	10	208	55(1-10)
16	10	217	55(11-20)
17	09	178	55(21-29)
18	09	526	55(30-31),138,145
19	07	191	63,147
20	07	191	135
21	10	532	53(1-10)
22	09	044	53(11-19)
23	09	072	53(20-28)
24	08	014	52
25	12	158	158,72
26	11	195	50,132(1-5)
27	10	048	132(6-10),51(1-5)
28	10	280	51(6-7),133,134
29	10	112	156
30	10	255	71
31	08	365	157
32	11	061	148

Table 6. Sizes, concentration parameters, and traverse numbers for local samples, Fording Domain 3 bedding. Numbers in parentheses indicate which observations from a traverse are included for split traverses.

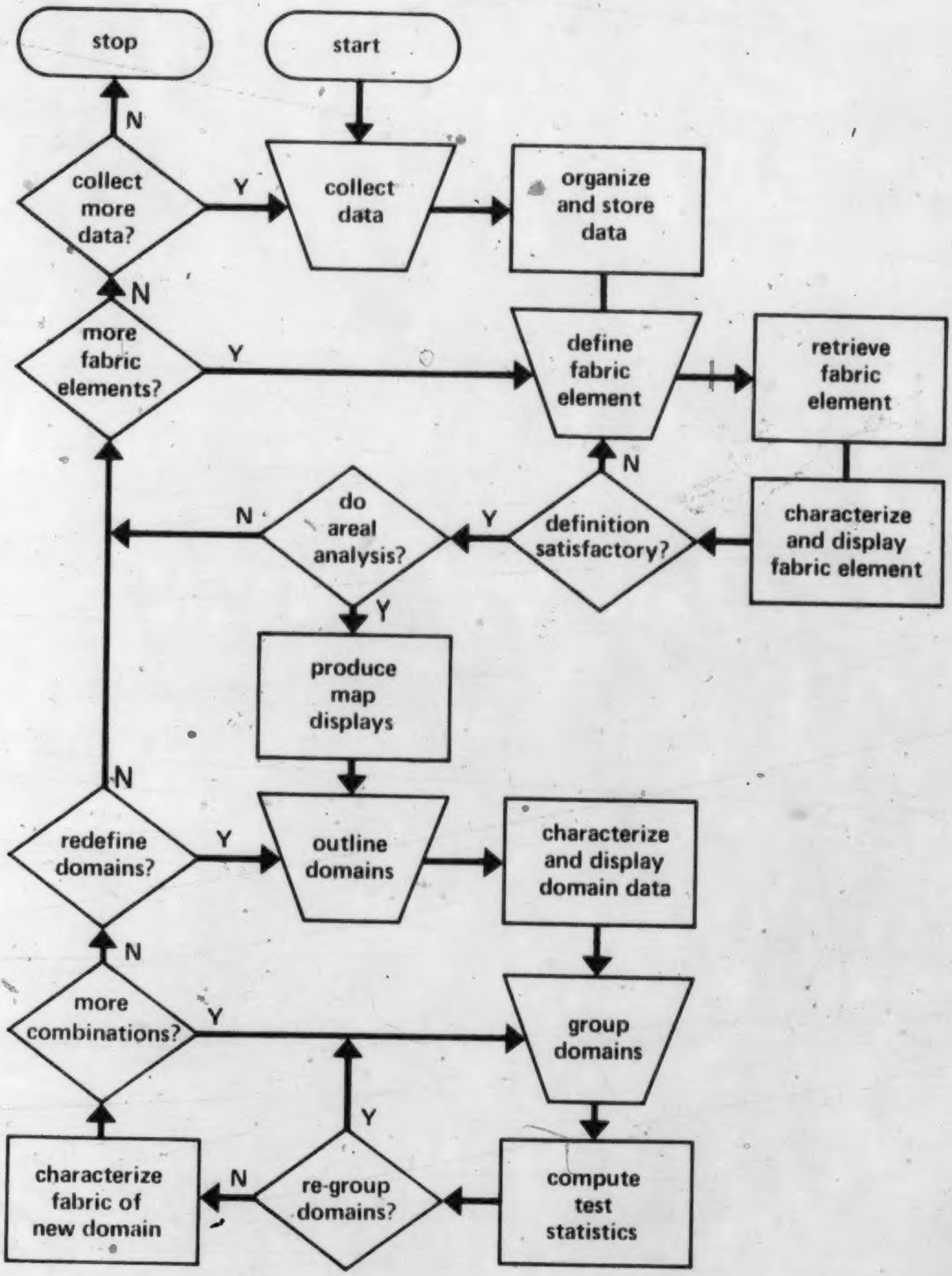

stop
start
N
collect more data?
Y
collect data
organize and store data
N
more fabric elements?
Y
define fabric element
retrieve fabric element
N
N
do areal analysis?
Y
definition satisfactory?
characterize and display fabric element
Y
produce map displays
N
redefine domains?
Y
outline domains
characterize and display domain data
N
more combinations?
Y
group domains
Y
characterize fabric of new domain
N
re-group domains?
compute test statistics

Figure 1. Procedure for Computer-Assisted Fabric Analysis

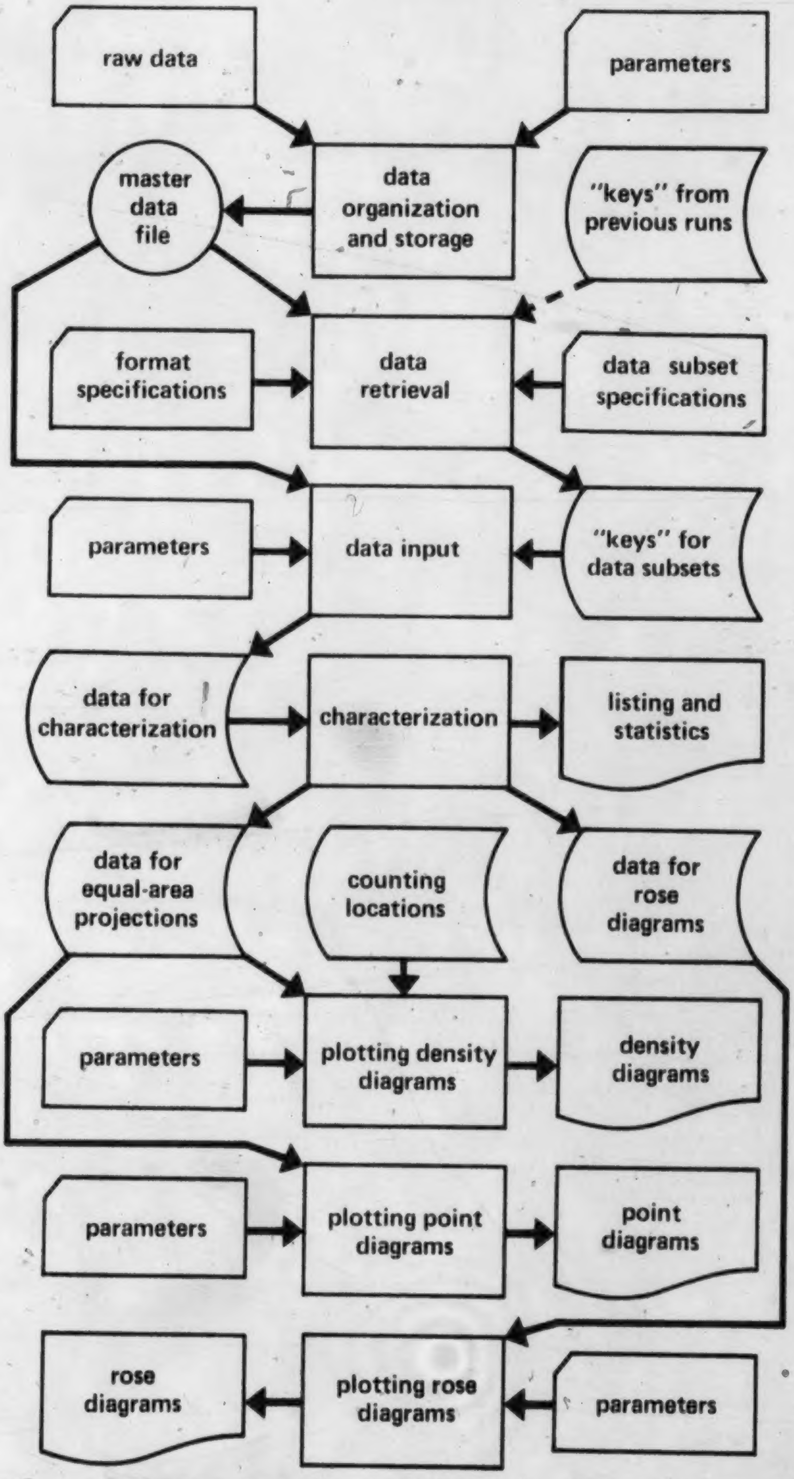

raw data
parameters
master data file
data organization and storage
"keys" from previous runs
format specifications
data retrieval
data subset specifications
parameters
data input
"keys" for data subsets
data for characterization
characterization
listing and statistics
data for equal-area projections
counting locations
data for rose diagrams
parameters
plotting density diagrams
density diagrams
parameters
plotting point diagrams
point diagrams
rose diagrams
plotting rose diagrams
parameters

Figure 2. Program Inputs and Outputs for Data Collection, Storage, Retrieval, Display and Characterization Operations

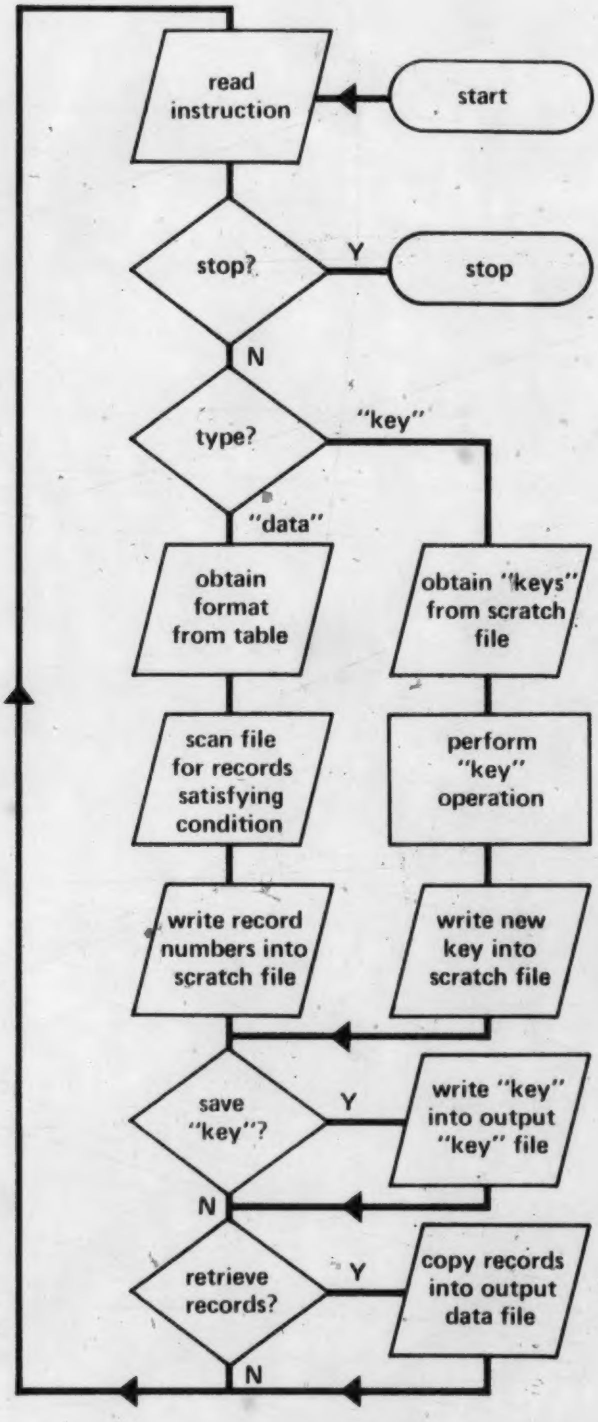

read instruction
start
stop?
Y
stop
N
type?
"key"
"data"
obtain format from table
obtain "keys" from scratch file
scan file for records satisfying condition
perform "key" operation
write record numbers into scratch file
write new key into scratch file
save "key"?
Y
write "key" into output "key" file
N
retrieve records?
Y
copy records into output data file
N

Figure 3. Simplified Flow Diagram for Program KEY4

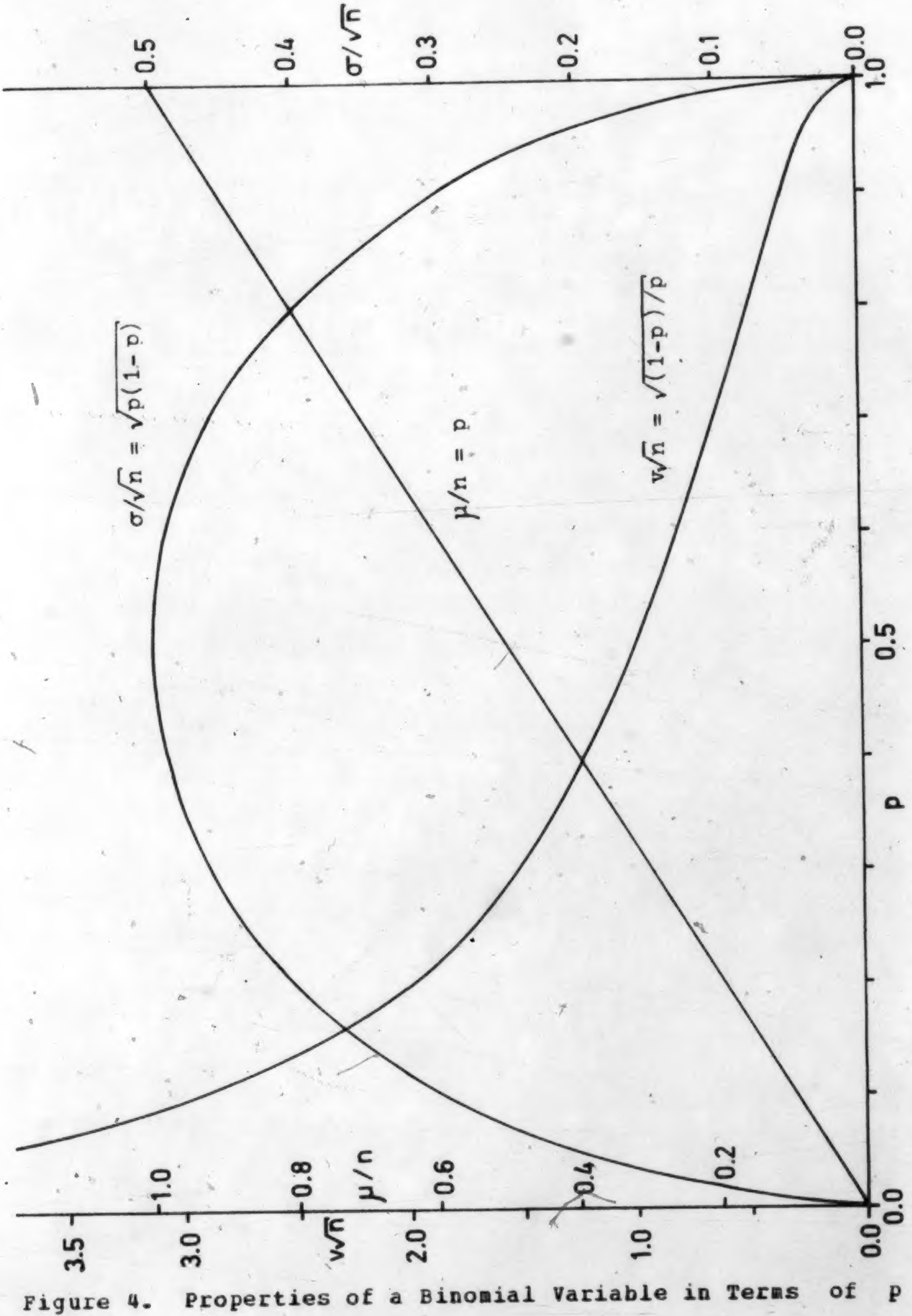


Figure 4. Properties of a Binomial Variable in Terms of p and n

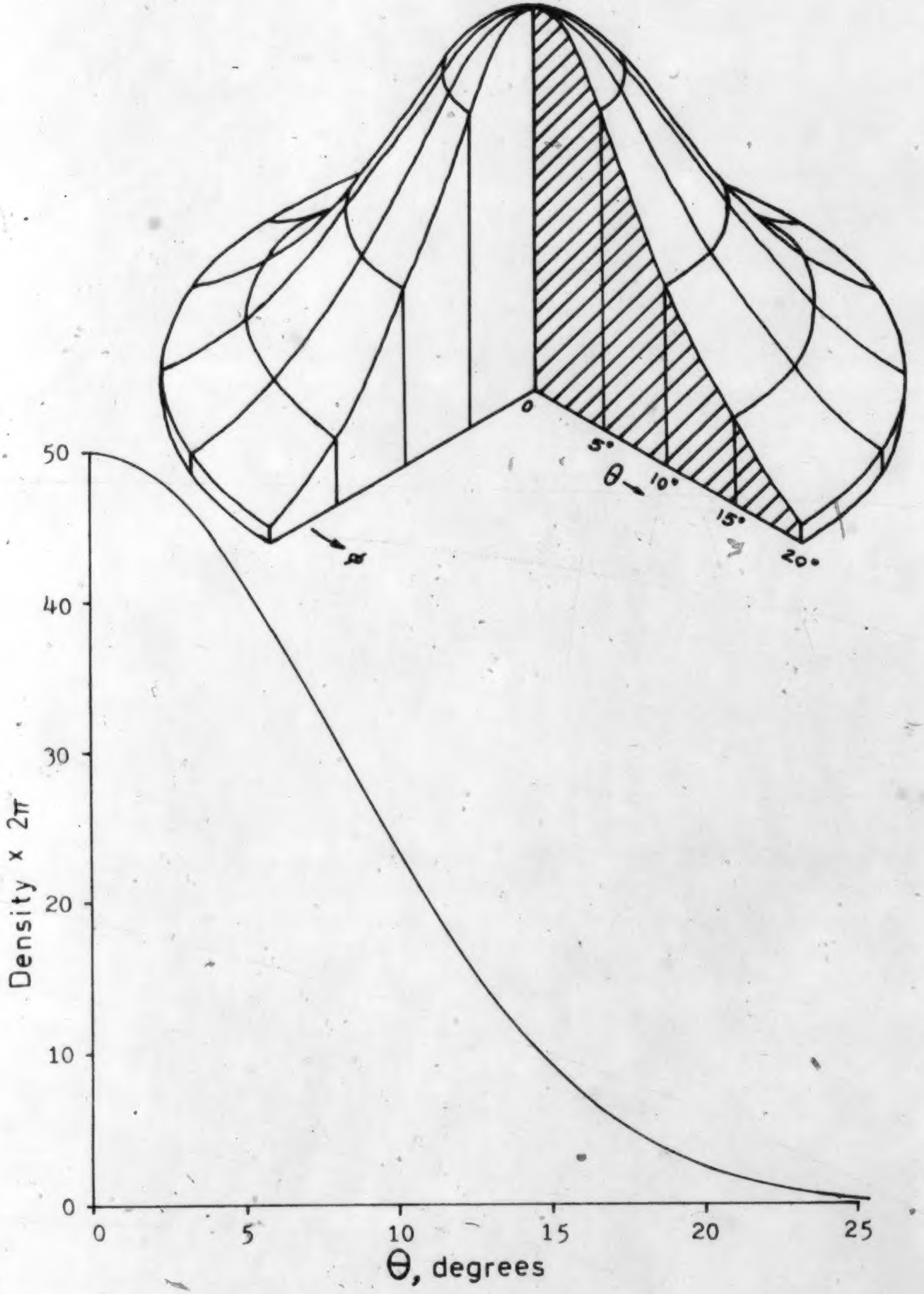


Figure 5. Fisher's Density Function for k=50

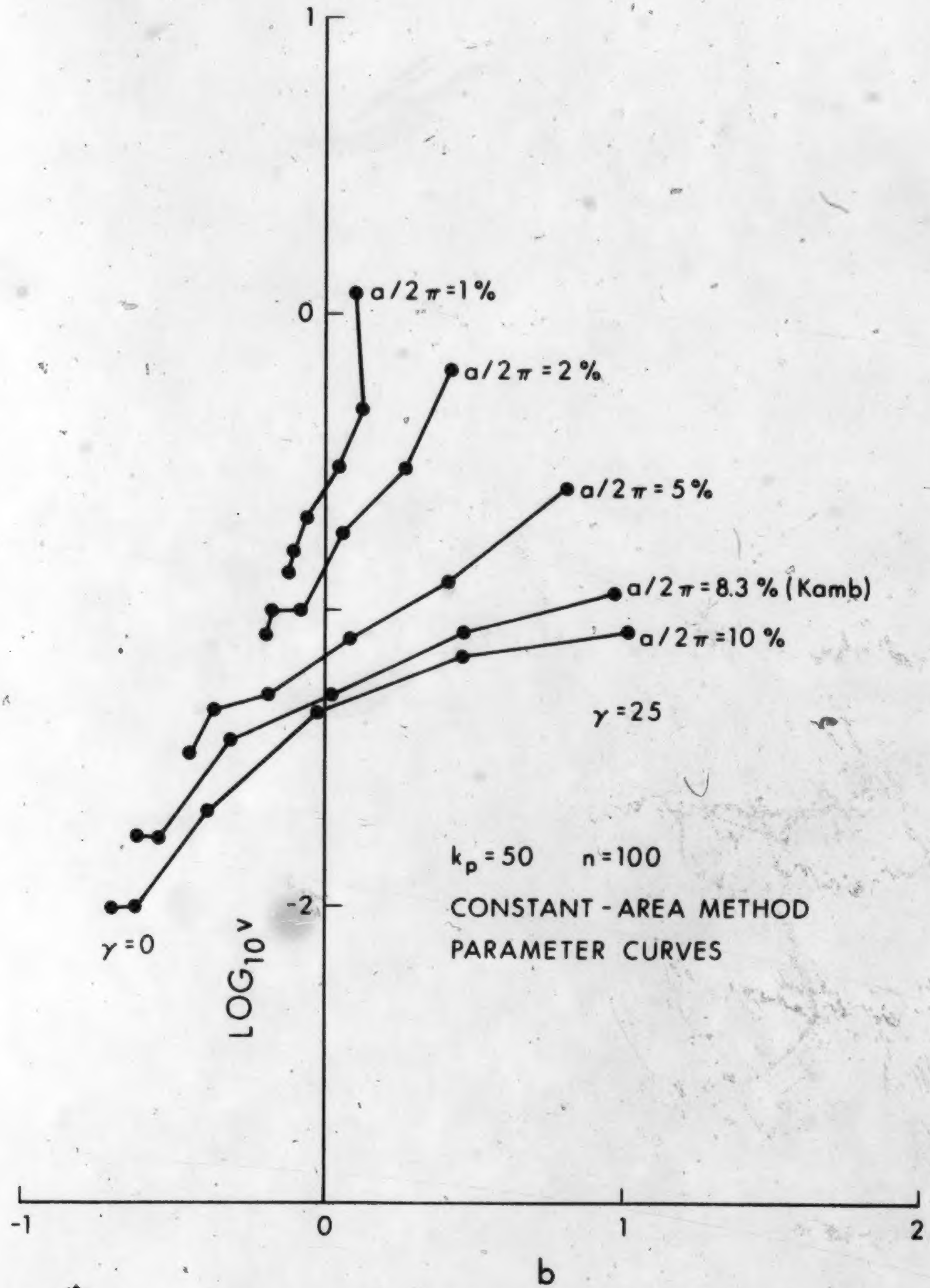


Figure 6. Parameter Curves for the Constant-Area Estimator

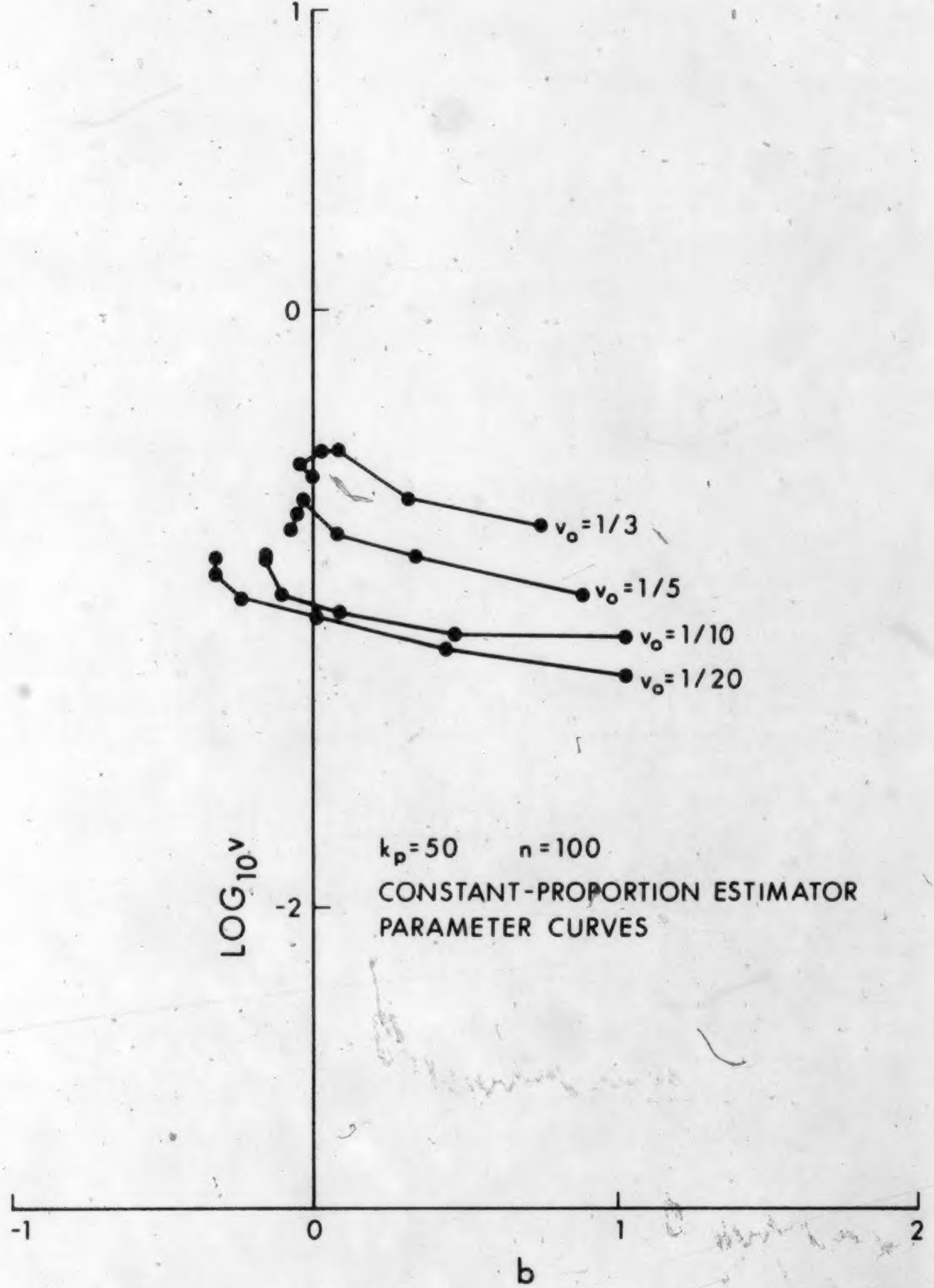


Figure 7. Parameter Curves for the Constant-Proportion Estimator

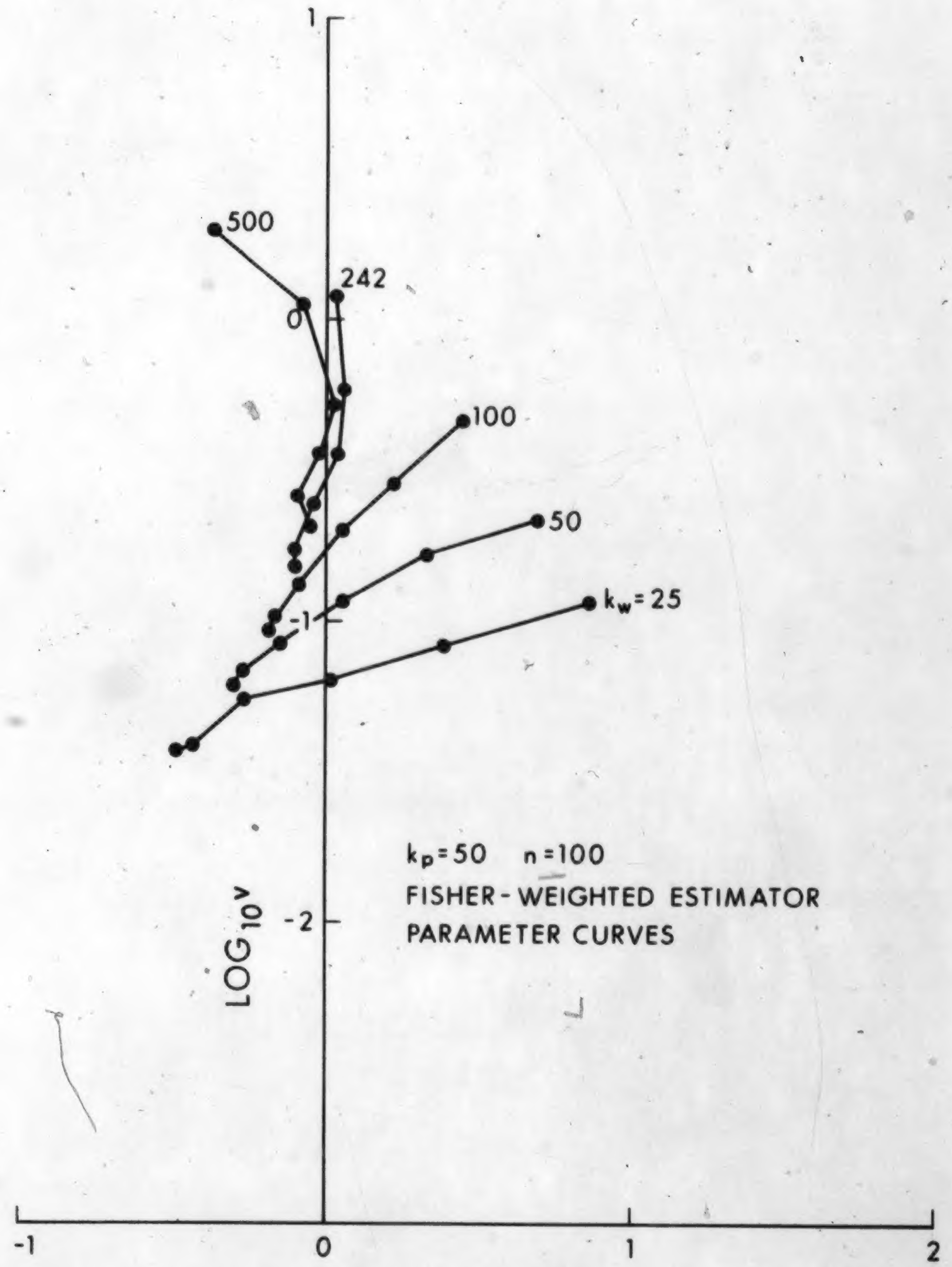


Figure 8. Parameter Curves for the Fisher-Weighted Estimator

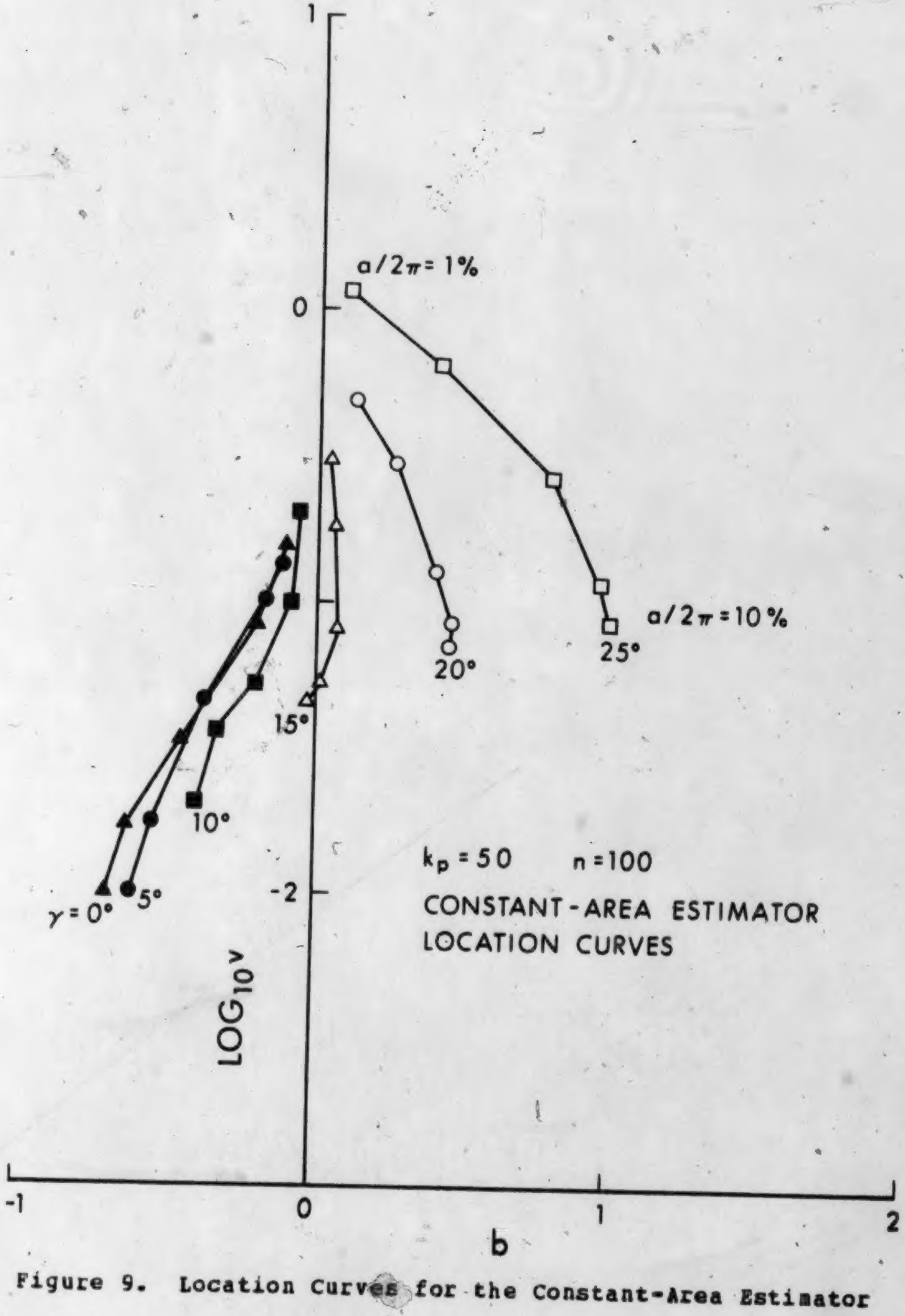


Figure 9. Location Curves for the Constant-Area Estimator

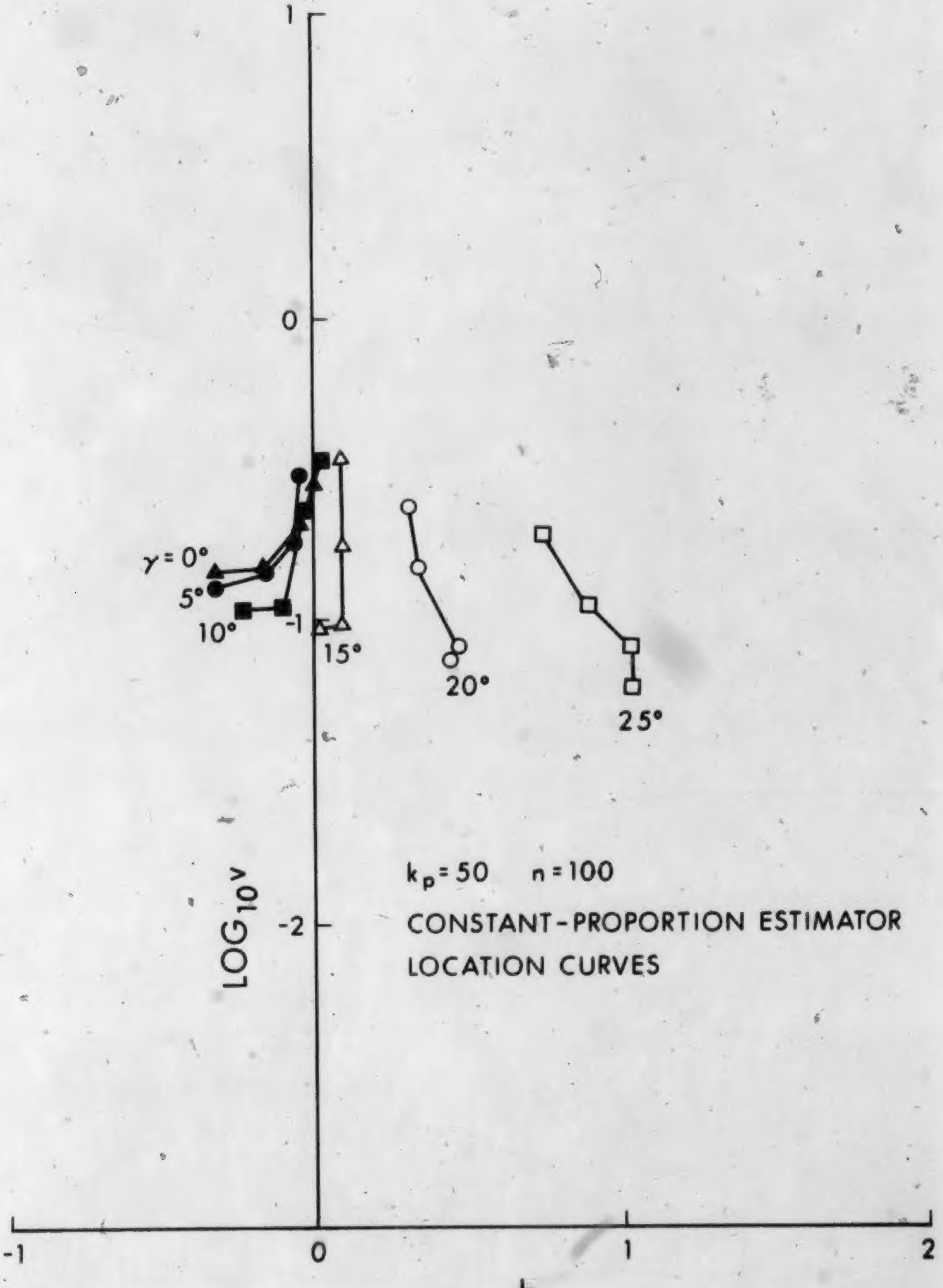


Figure 10. Location Curves for the Constant-Proportion Estimator

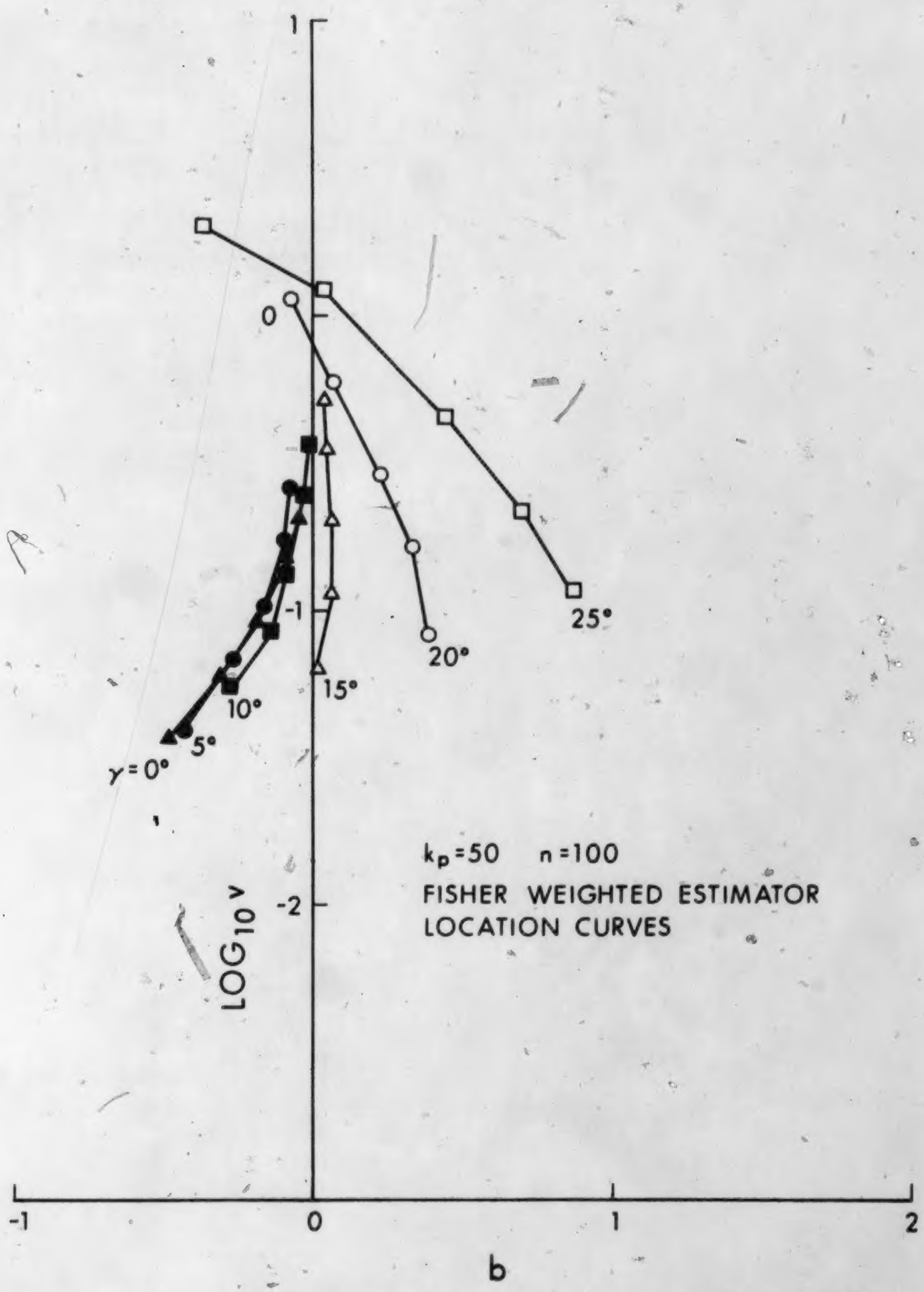


Figure 11. Location Curves for the Fisher-Weighted Estimator

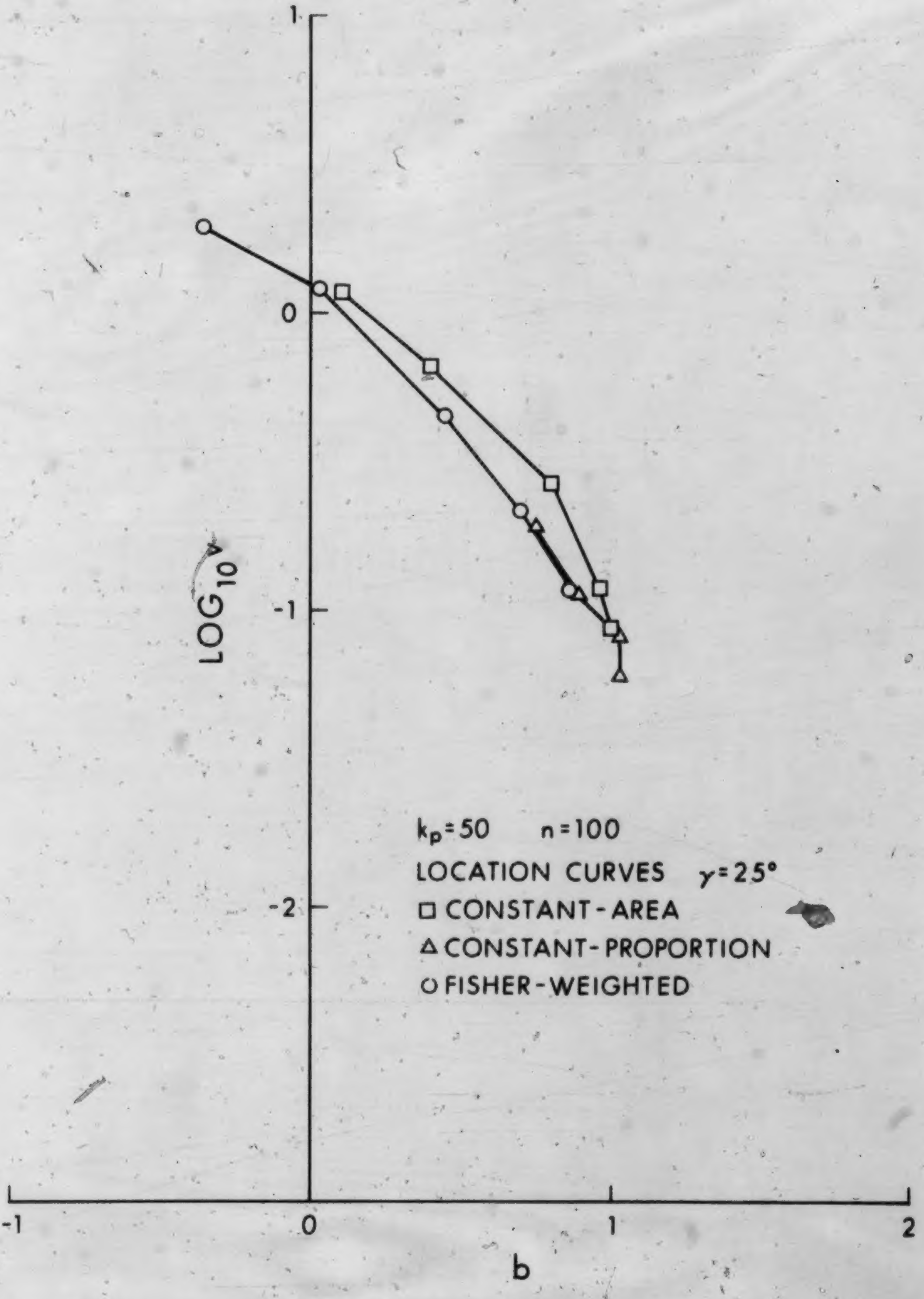


Figure 12. Location Curves for Gamma = 25 degrees

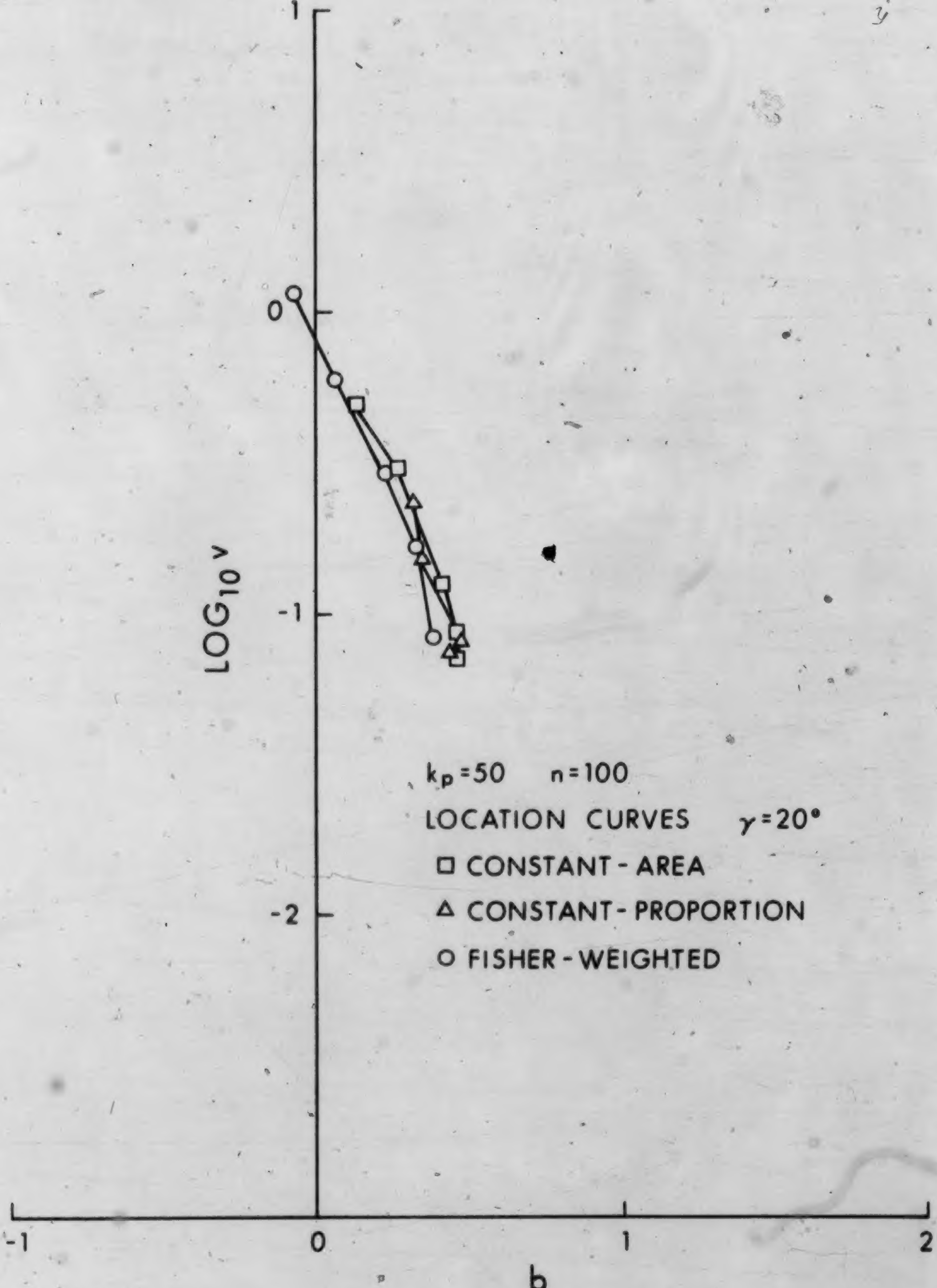


Figure 13. Location Curves for Gamma = 20 degrees

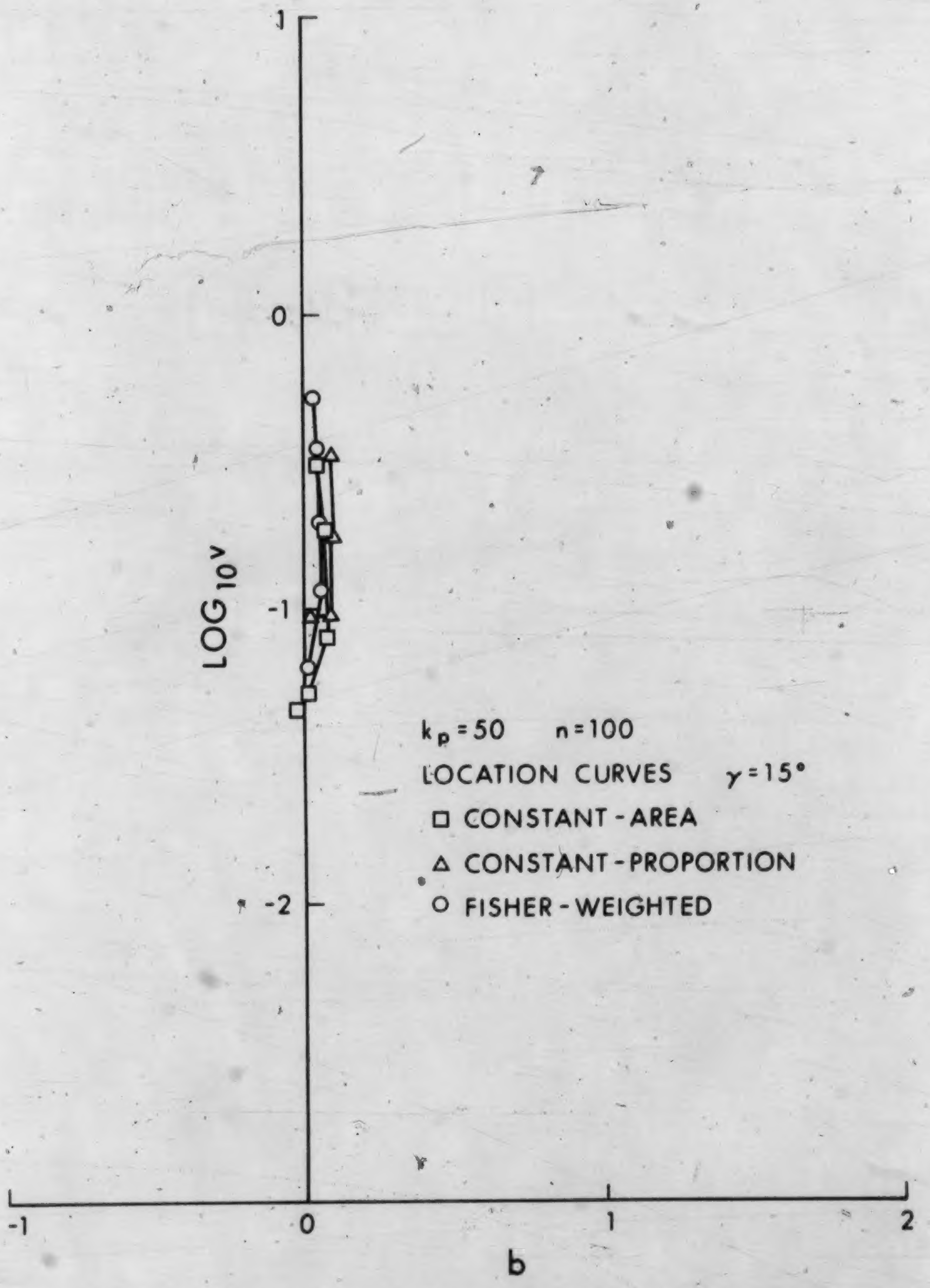


Figure 14. Location Curves for Gamma = 15 degrees

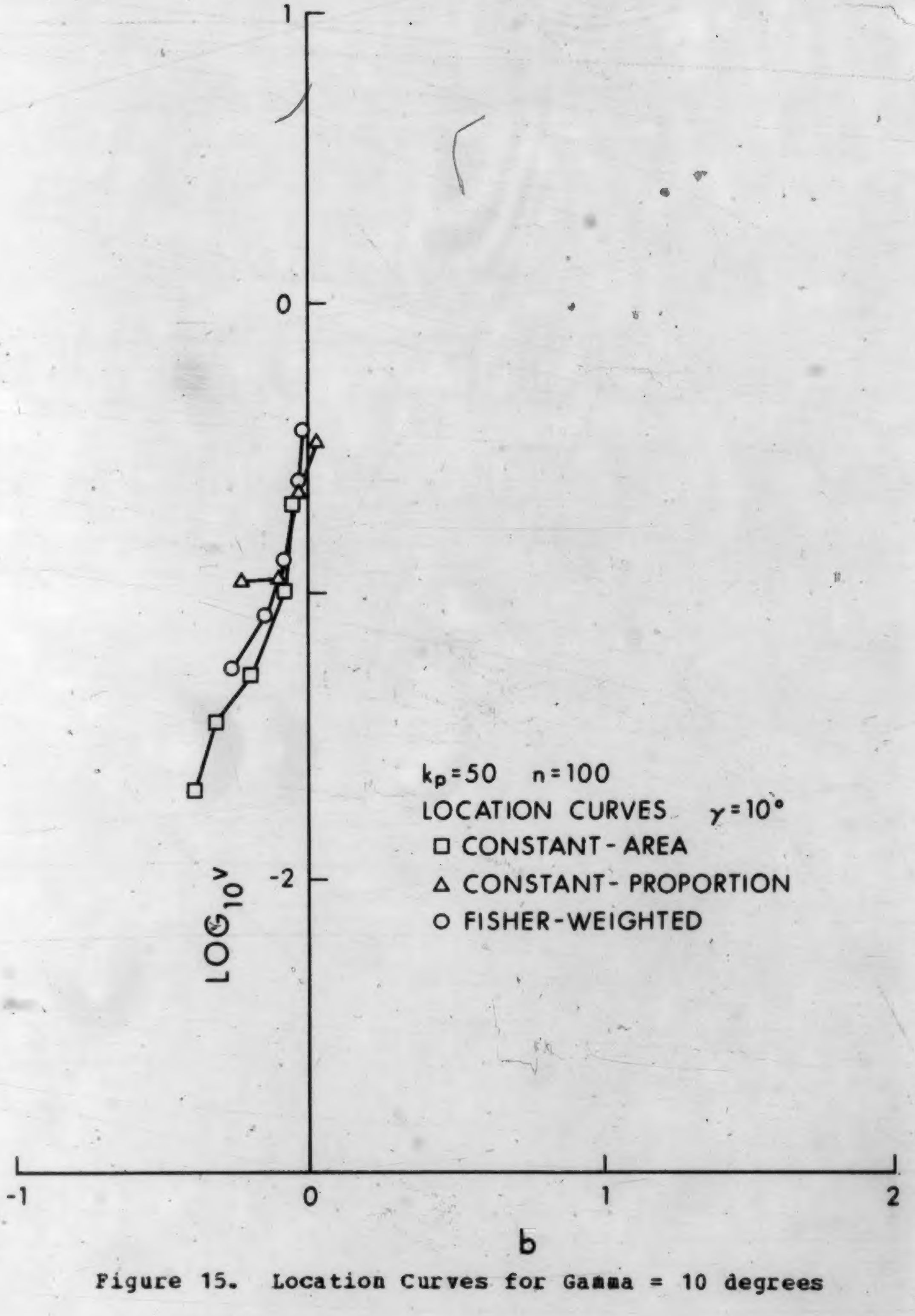


Figure 15. Location Curves for Gamma = 10 degrees

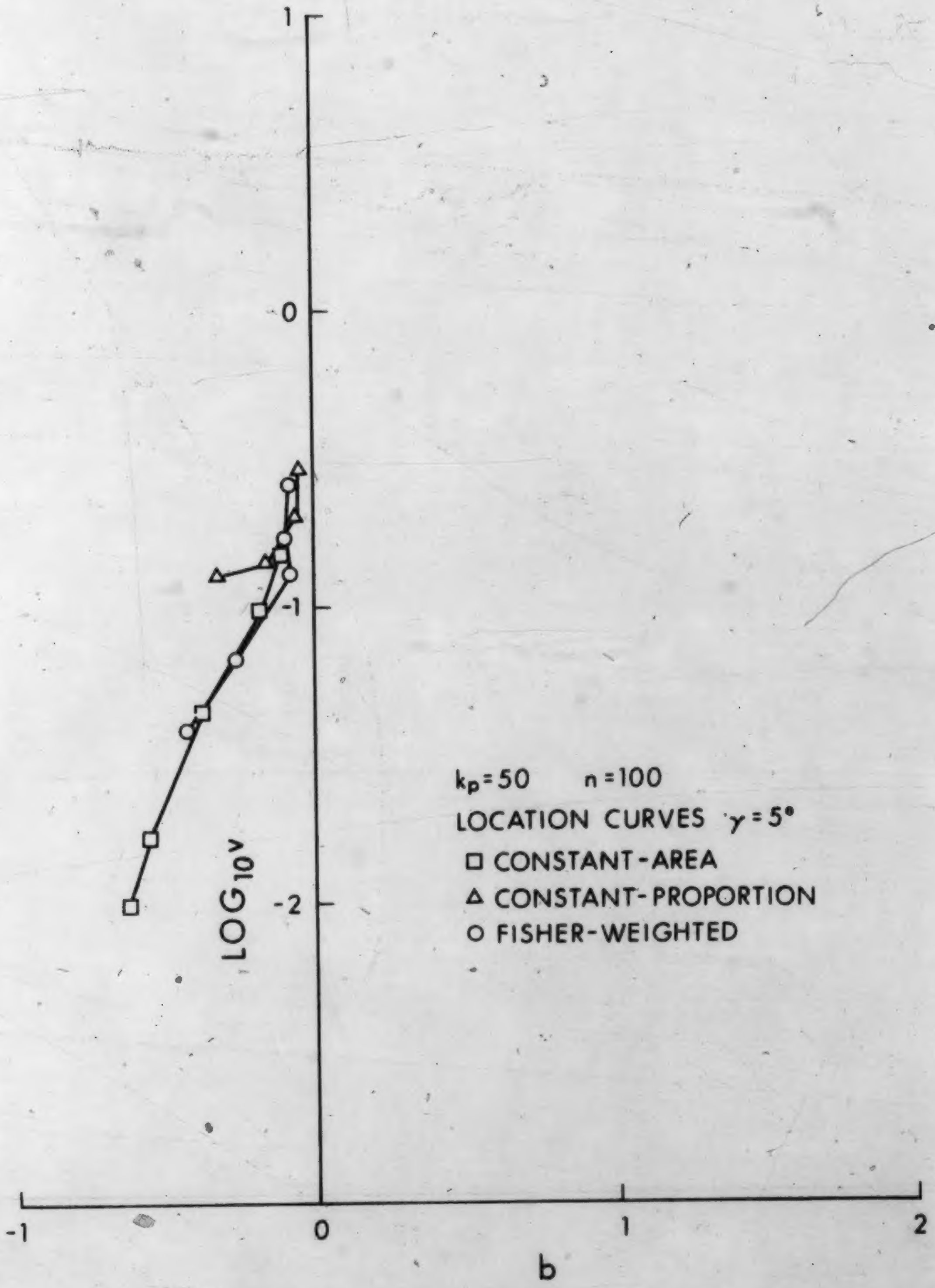


Figure 16. Location Curves for Gamma = 5 degrees

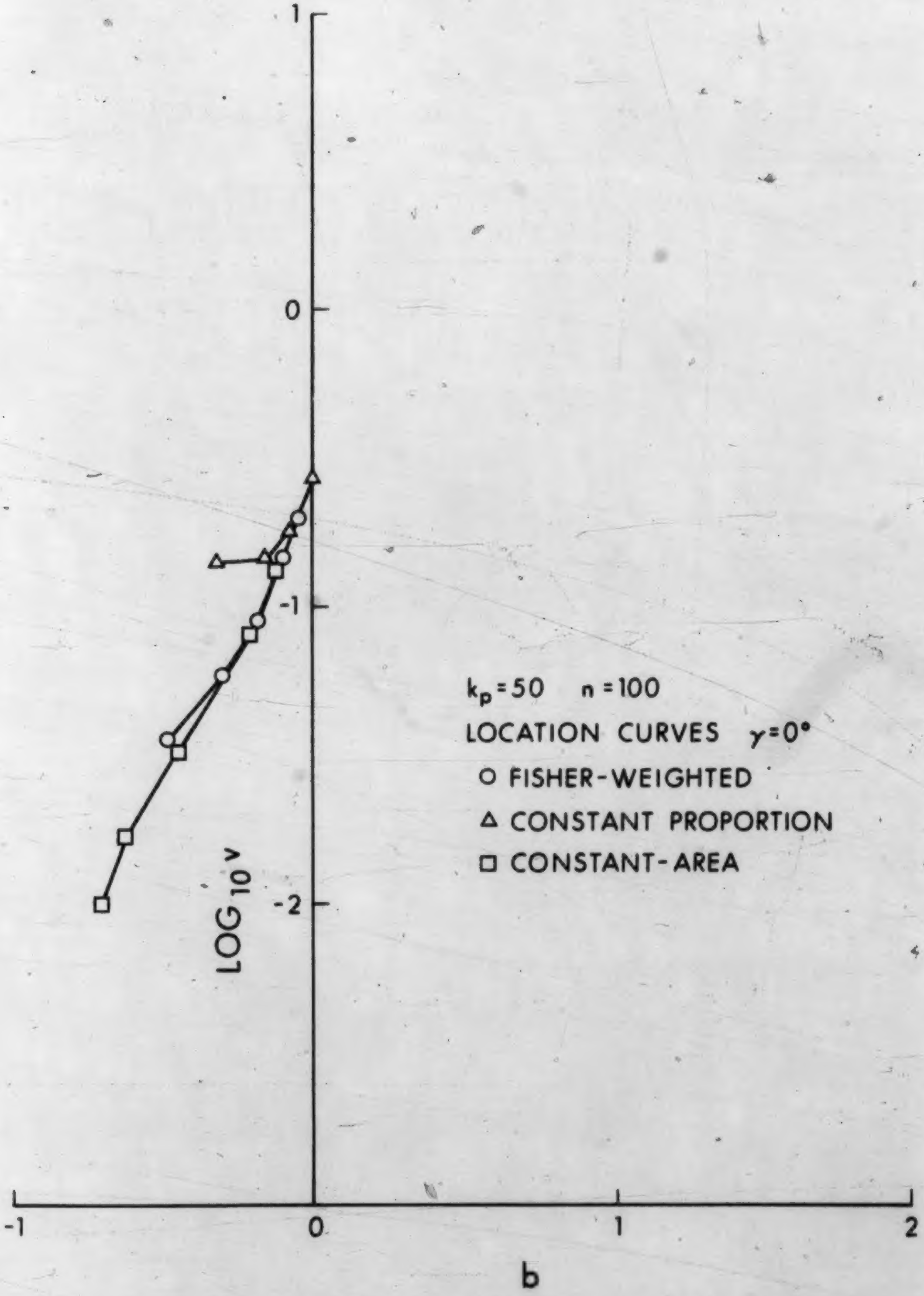


Figure 17. Location Curves for Gamma = 0 degrees

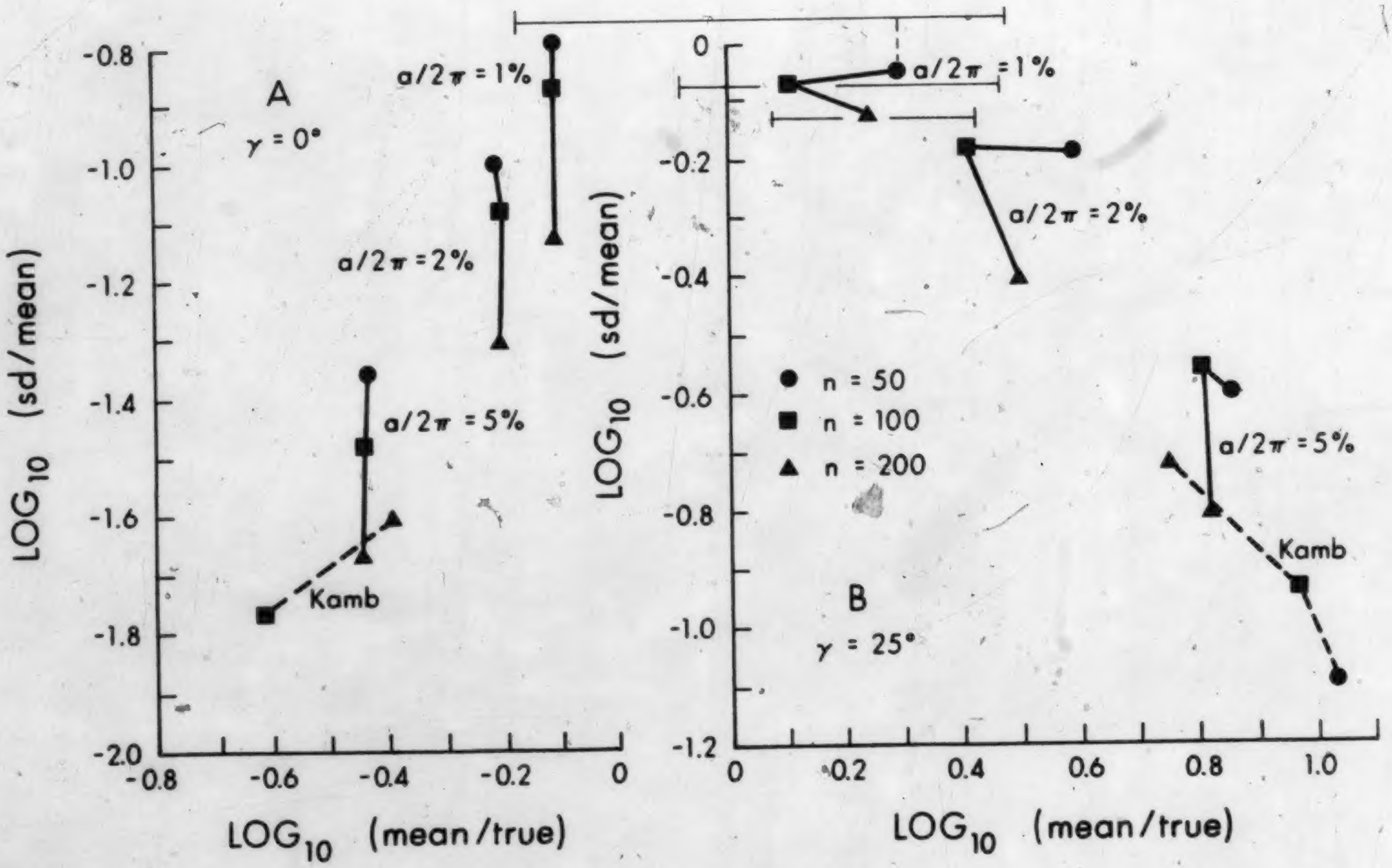


Figure 18. Effect of Sample Size on the Constant-Area Estimator

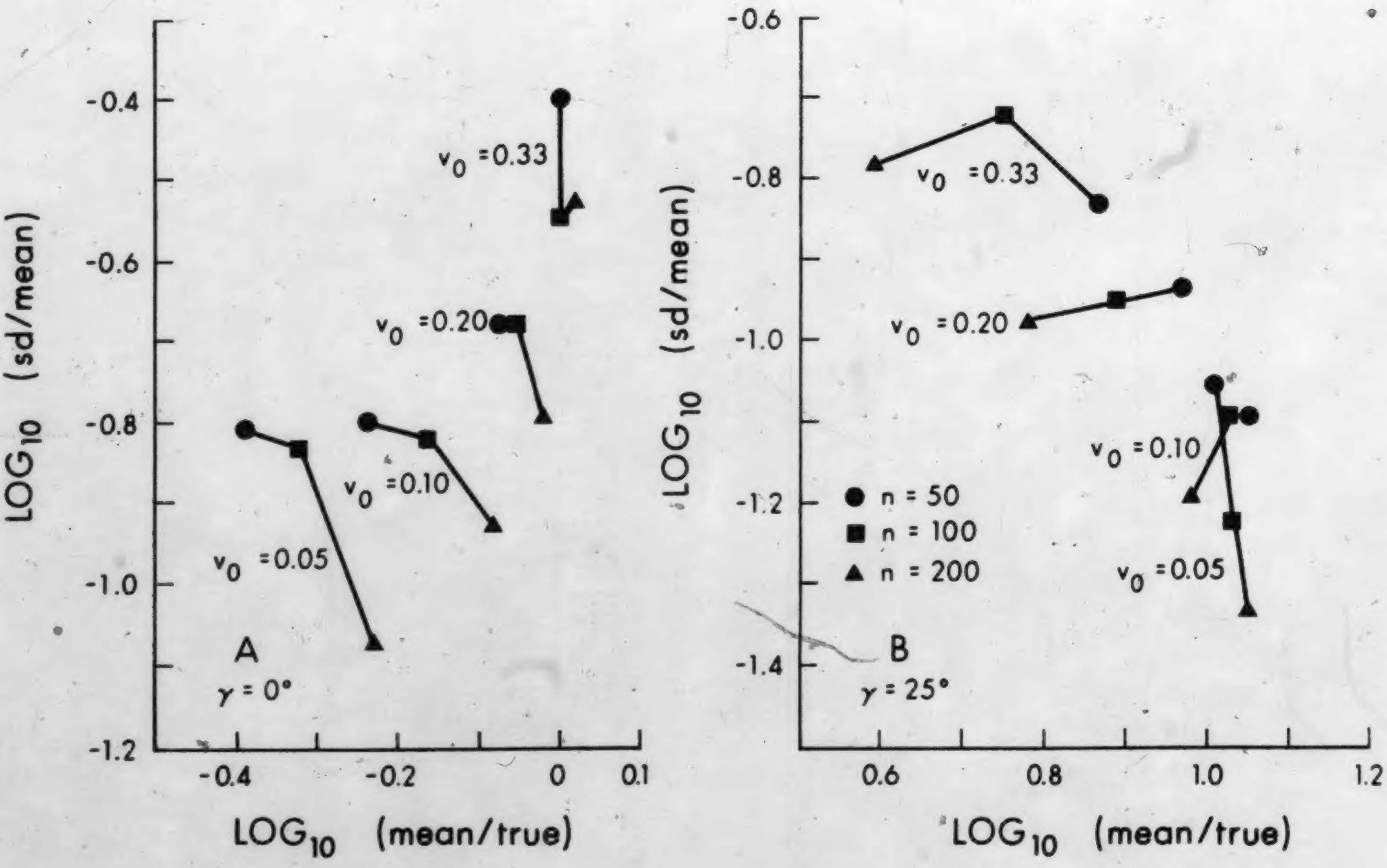


Figure 19. Effect of Sample Size on the Constant-Proportion Estimator

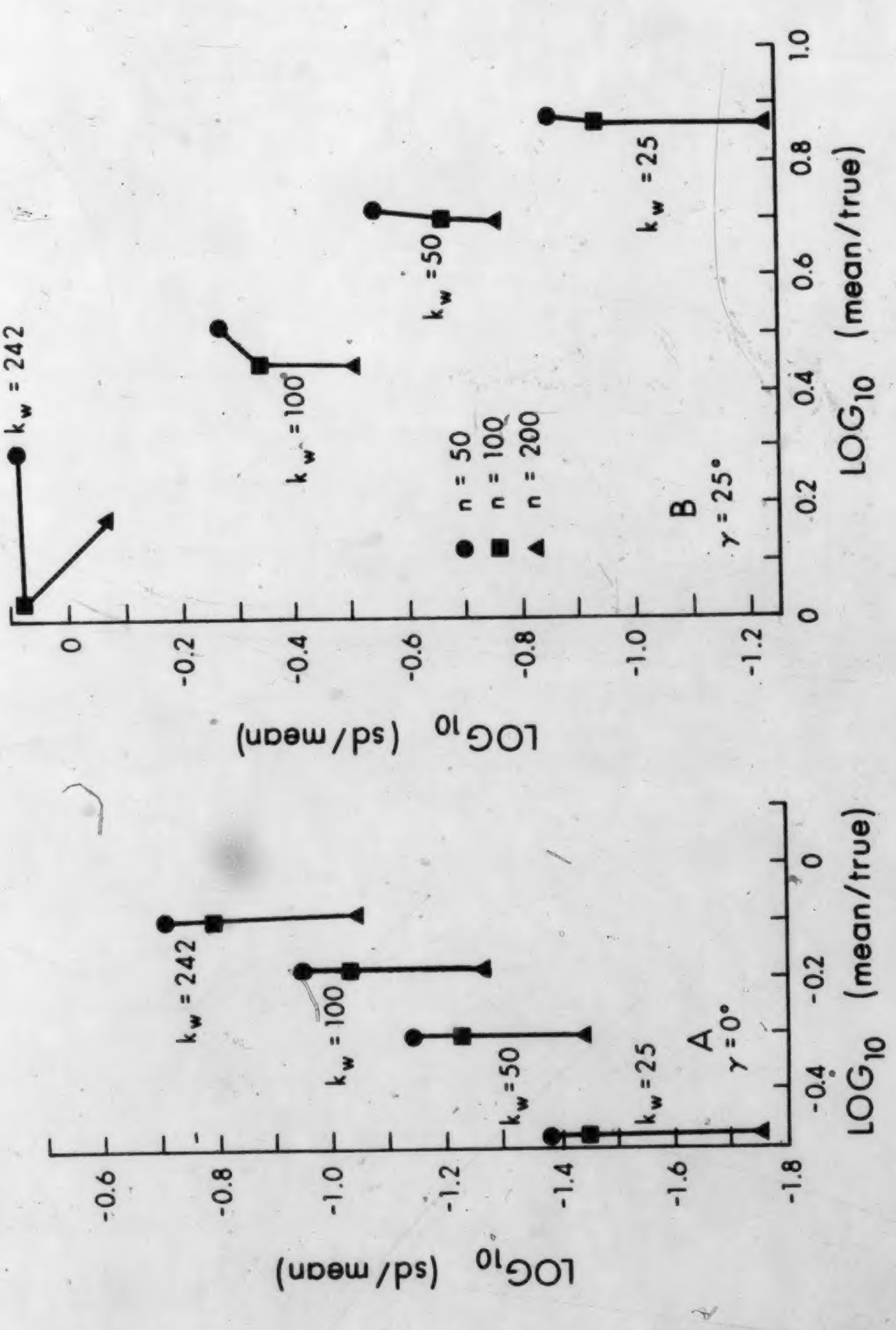


Figure 20. Effect of Sample Size on the Fisher-Weighted Estimator

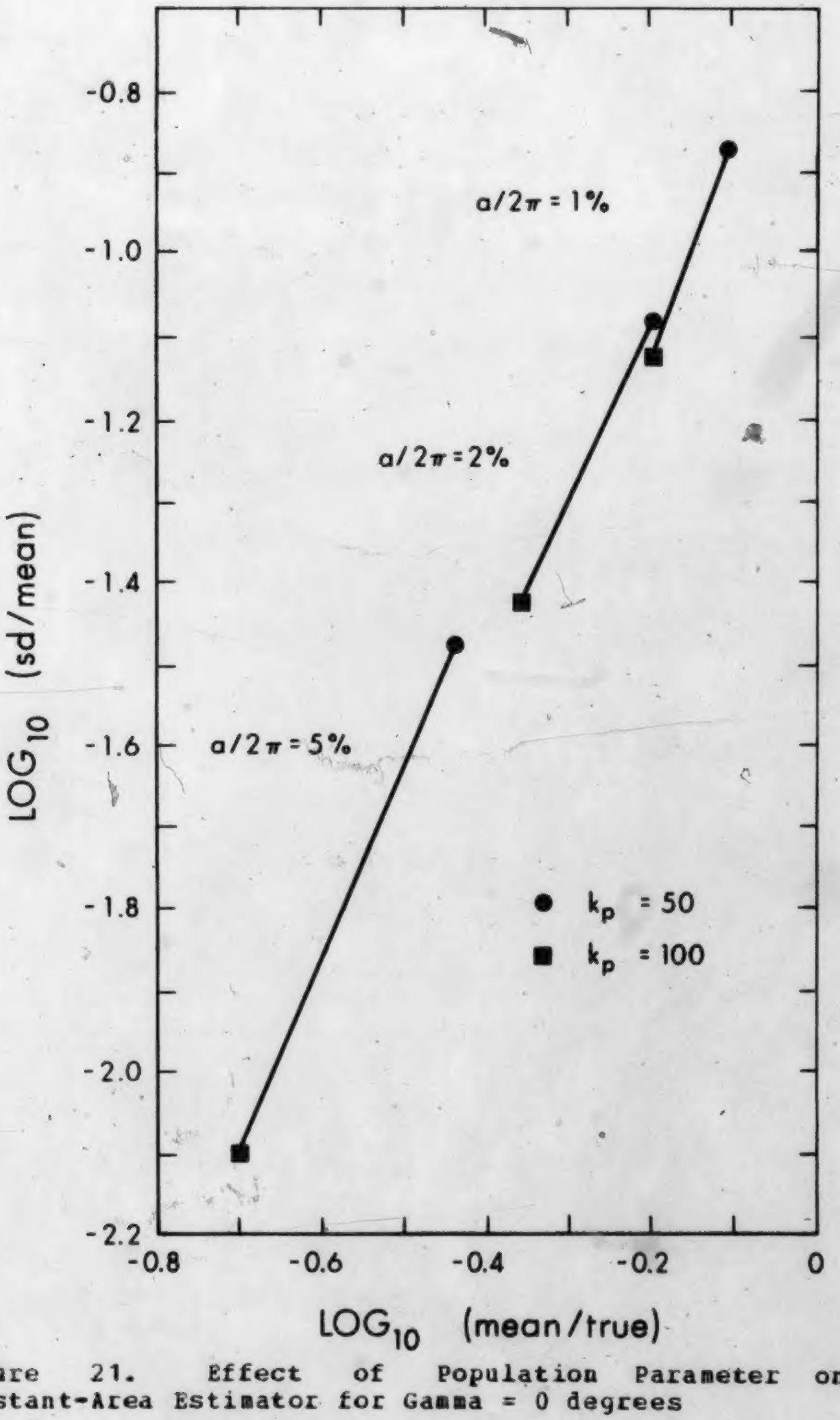


Figure 21. Effect of Population Parameter on the Constant-Area Estimator for Gamma = 0 degrees

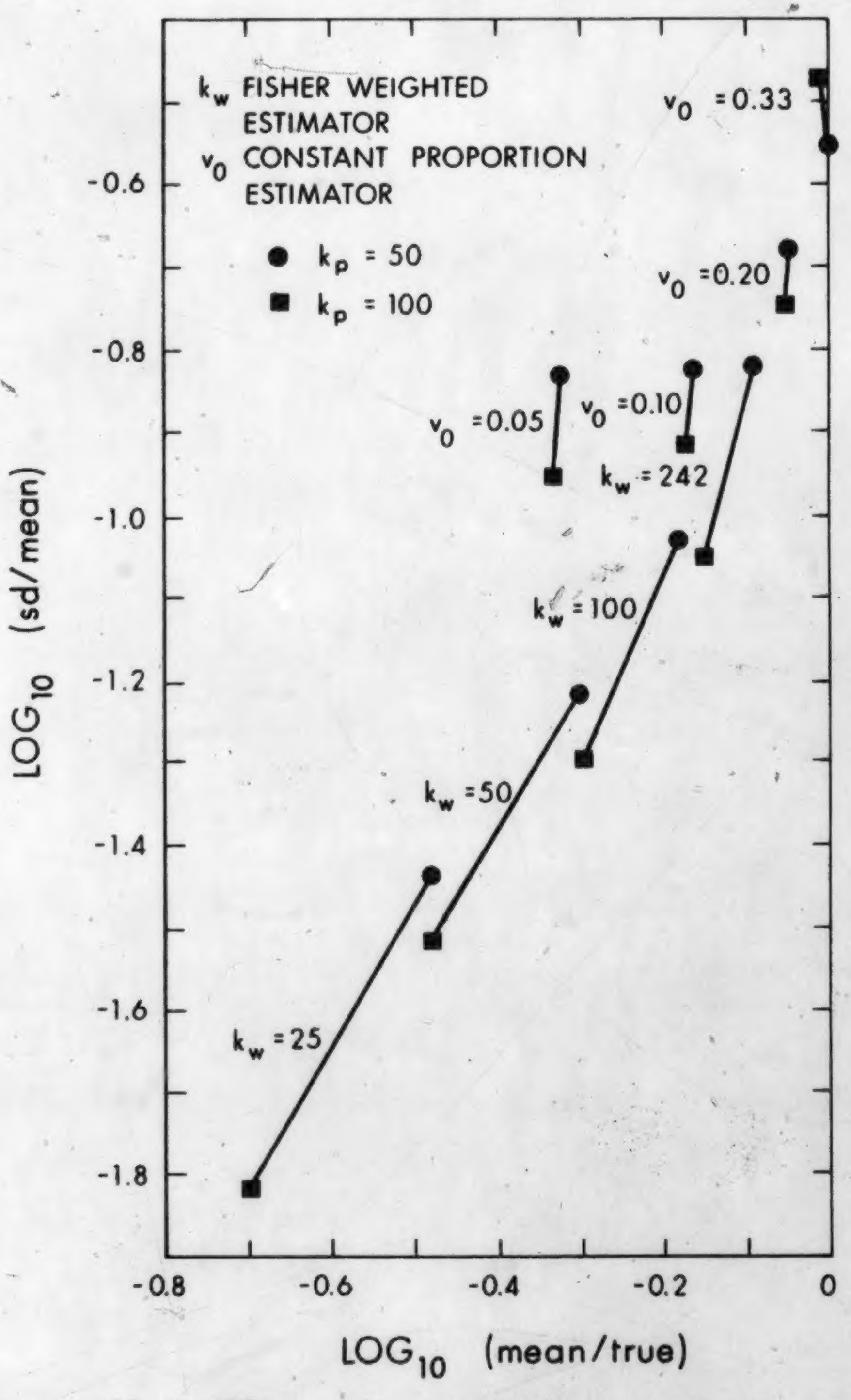


Figure 22. Effect of Population Parameter on the Constant-Proportion Fisher-Weighted Estimators for Gamma = 0 degrees

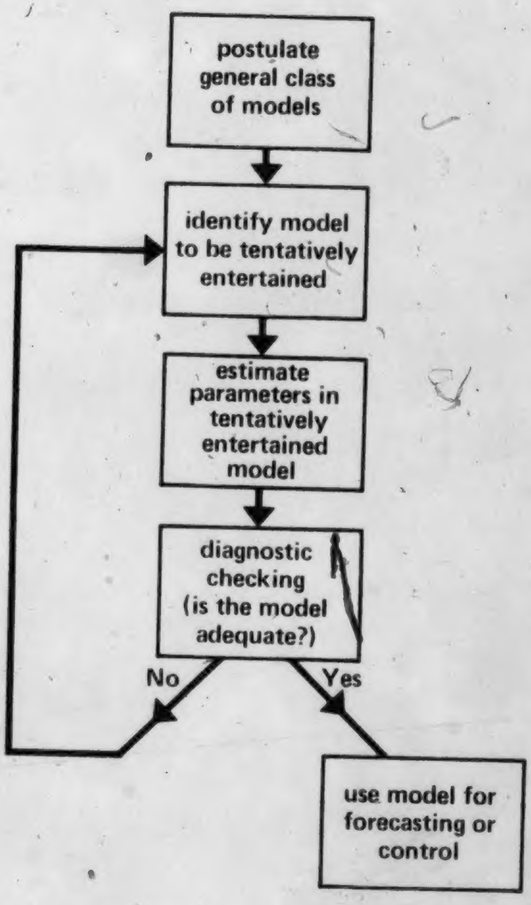
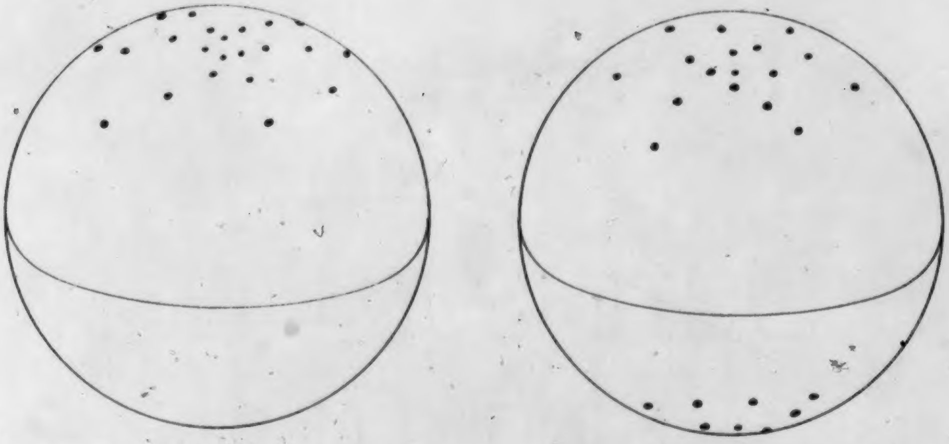
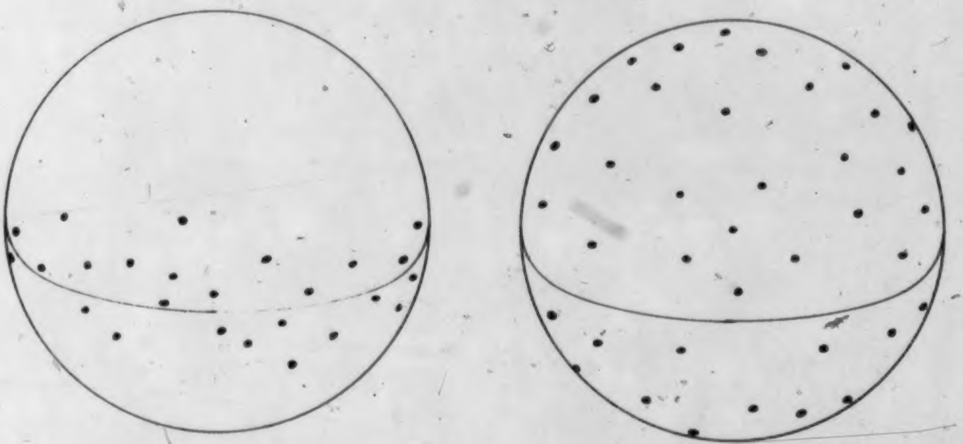

postulate general class of models
identify model to be tentatively entertained
estimate parameters in tentatively entertained model
diagnostic checking (is the model adequate?)
No
Yes
use model for forecasting or control

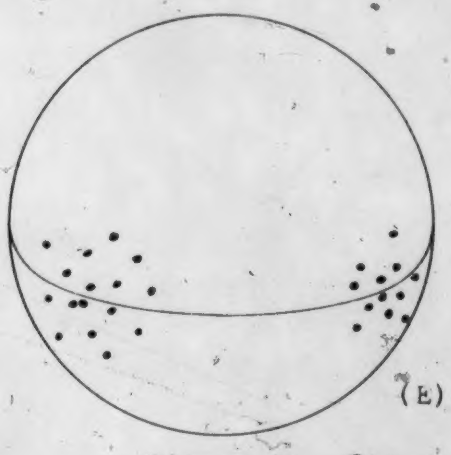
Figure 23. The Model-Building Process



(A) polar (B) bipolar



(C) equatorial or girdle (D) uniform



(E) mixed

Figure 24. Spherical Point Distributions

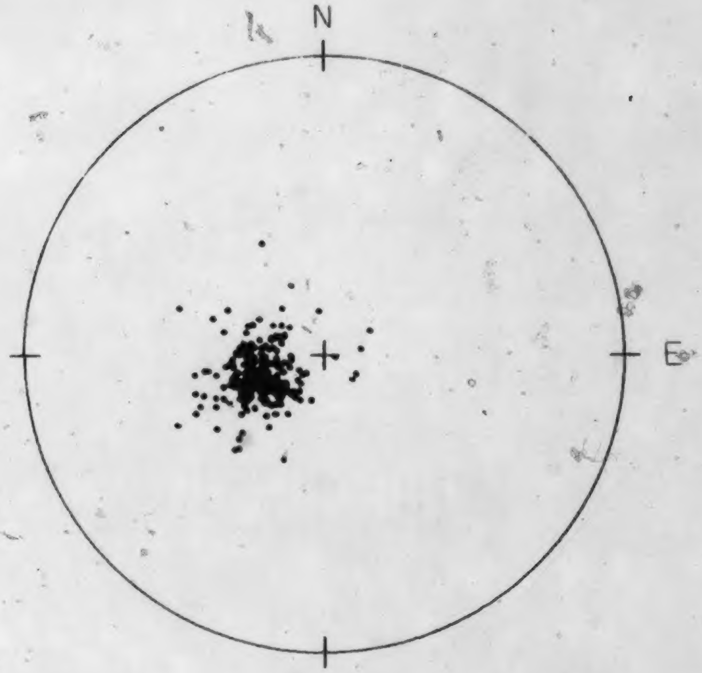


Figure 25. Lower-Hemisphere Equal-Area Projection of 297 Poles to Bedding in Domain 3 at the Fording Coal Mine

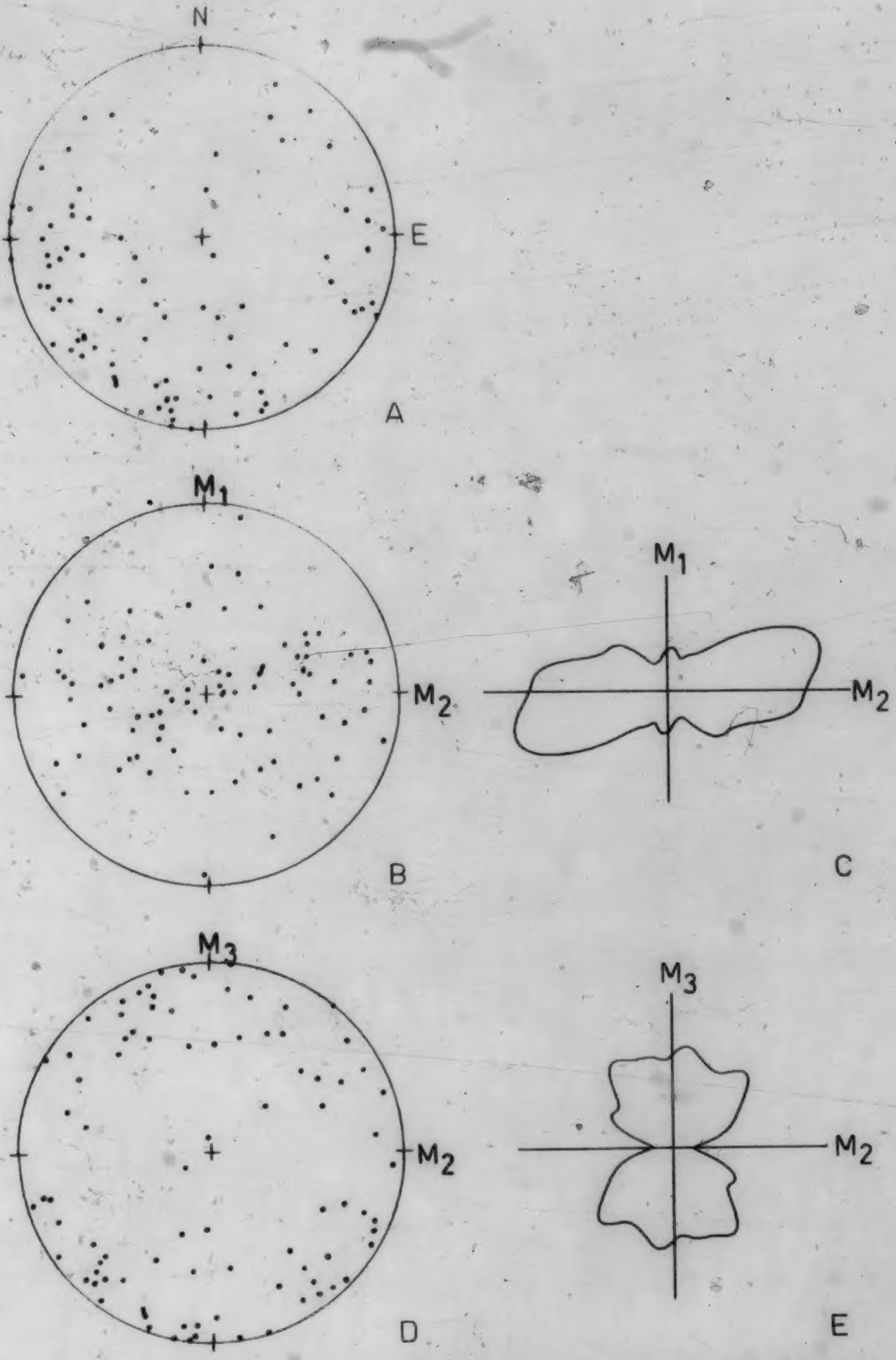


Figure 26. Displays for Sample 30; 90 observations

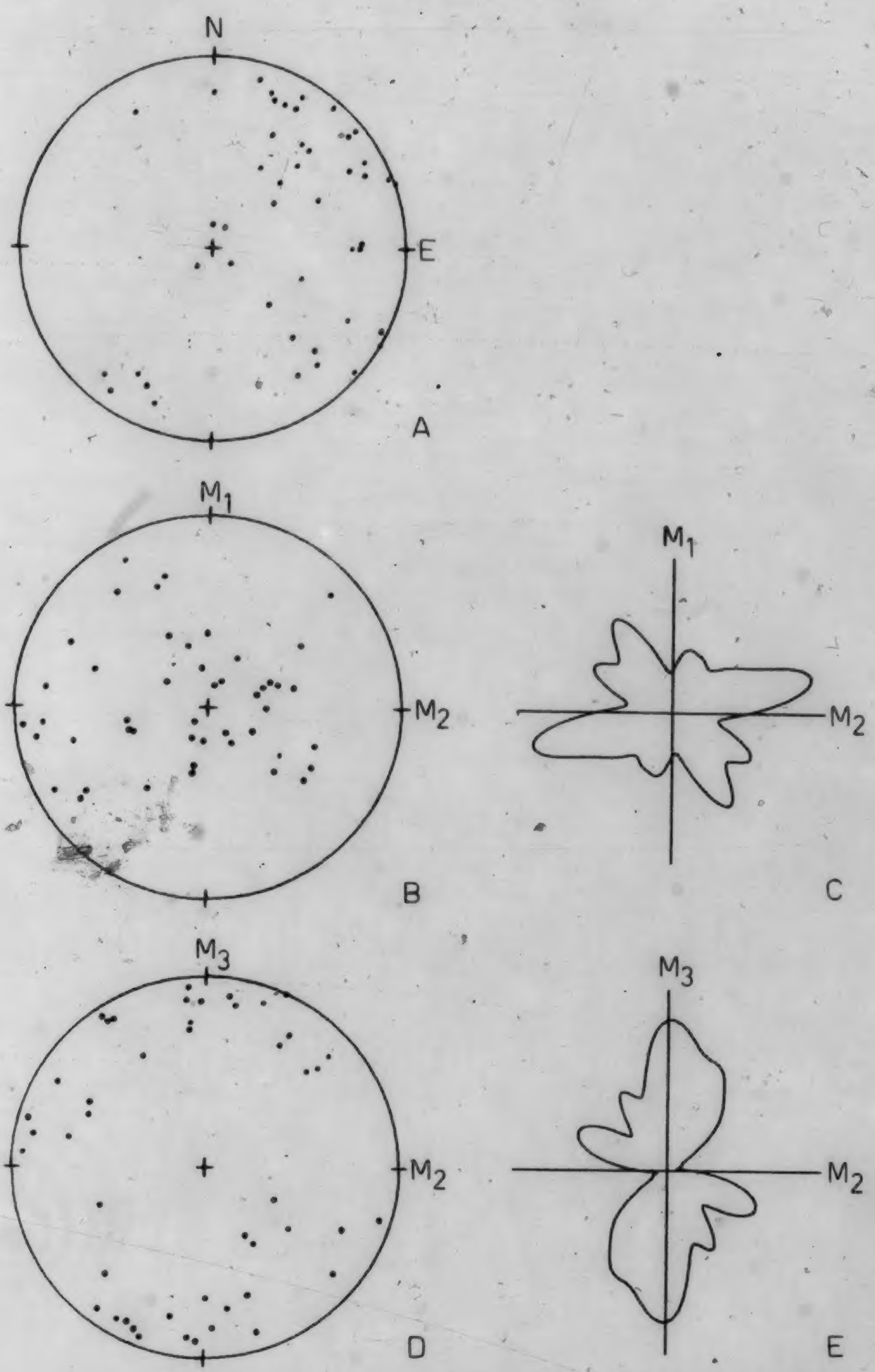


Figure 27. Displays for Sample 100; 49 observations

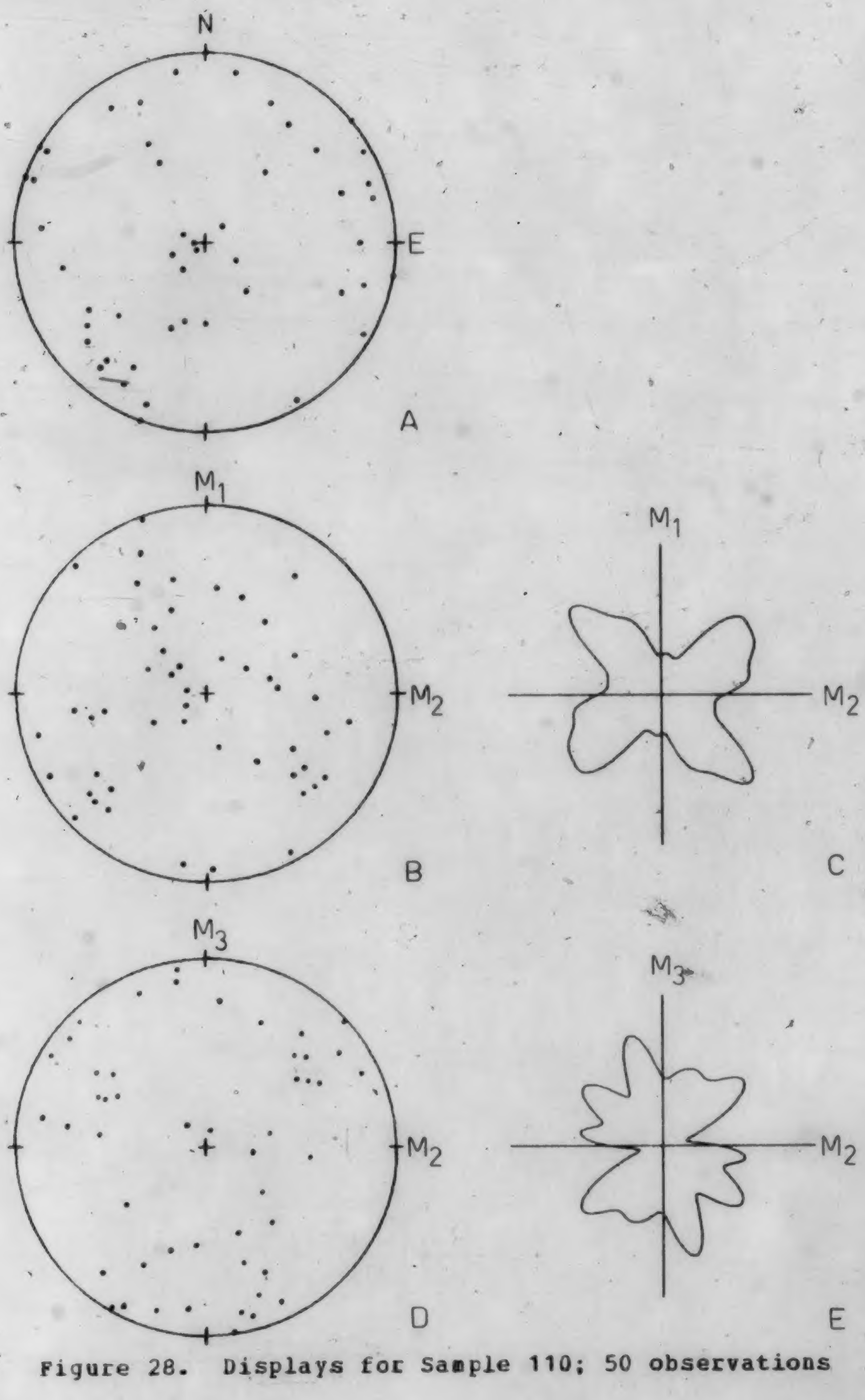


Figure 28. Displays for Sample 110; 50 observations

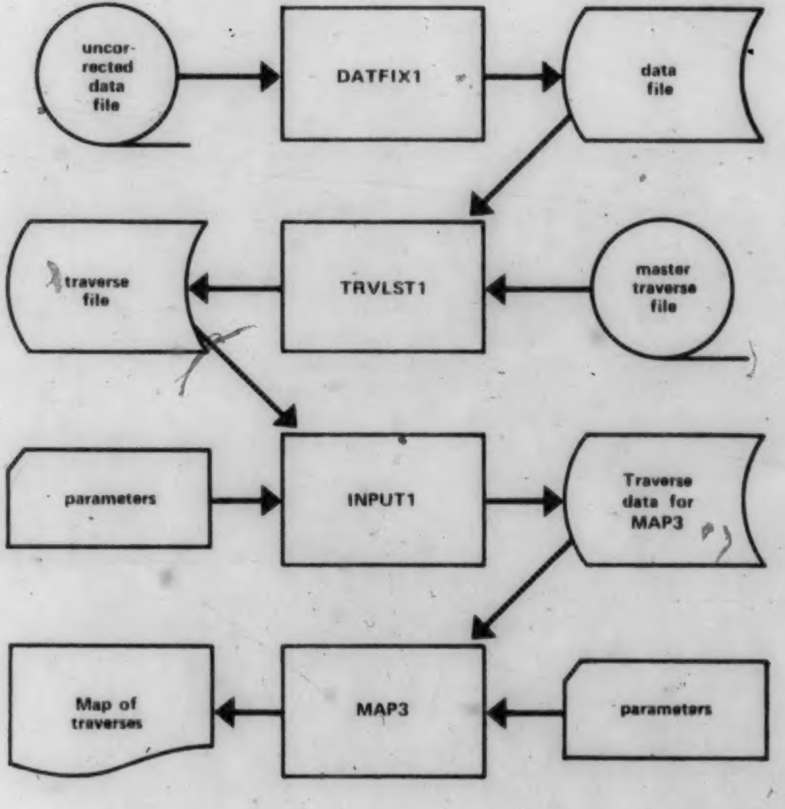

uncor-
rected
data
file
DATFIX1
data
file
traverse
file
TRVLST1
master
traverse
file
parameters
INPUT1
Traverse
data for
MAP3
Map of
traverses
MAP3
parameters

Figure 29. Procedure for Areal Analysis - Part 1

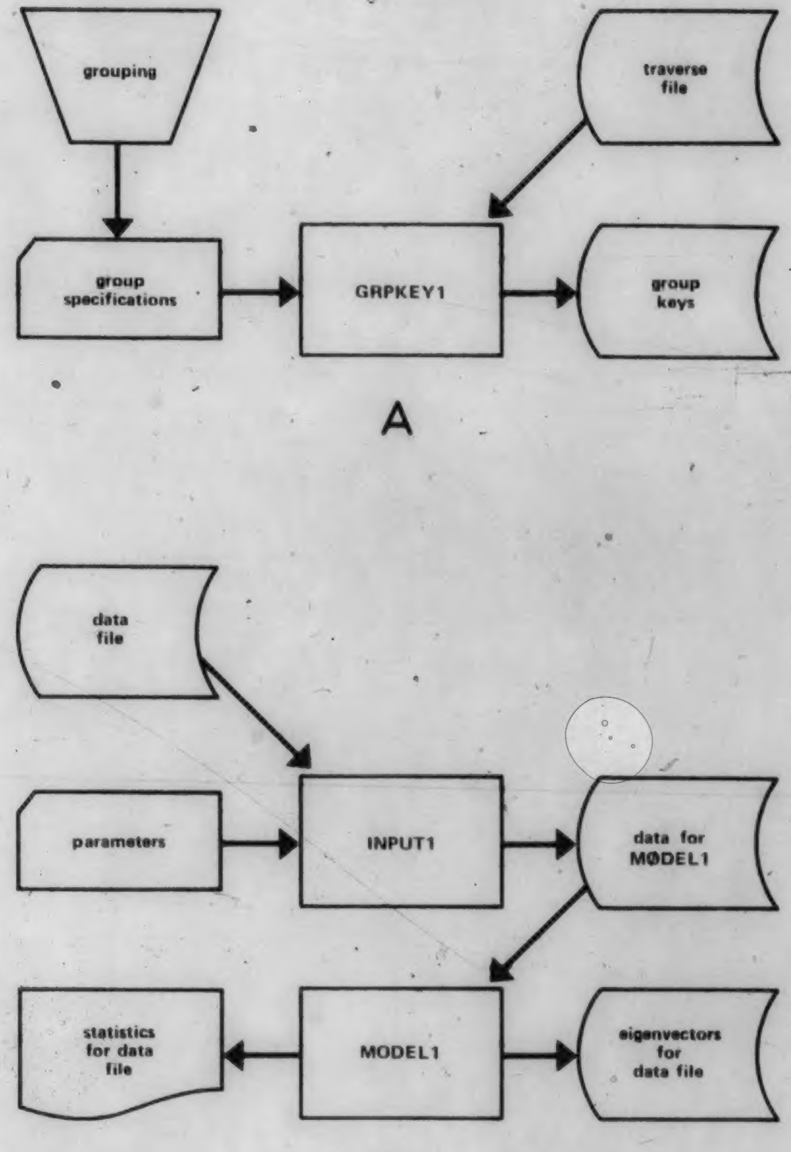

grouping
traverse file
group specifications
GRPKEY1
group keys
A
data file
parameters
INPUT1
data for MØDEL1
statistics for data file
MODEL1
eigenvectors for data file
B

Figure 30. Procedure for Areal Analysis - Part 2

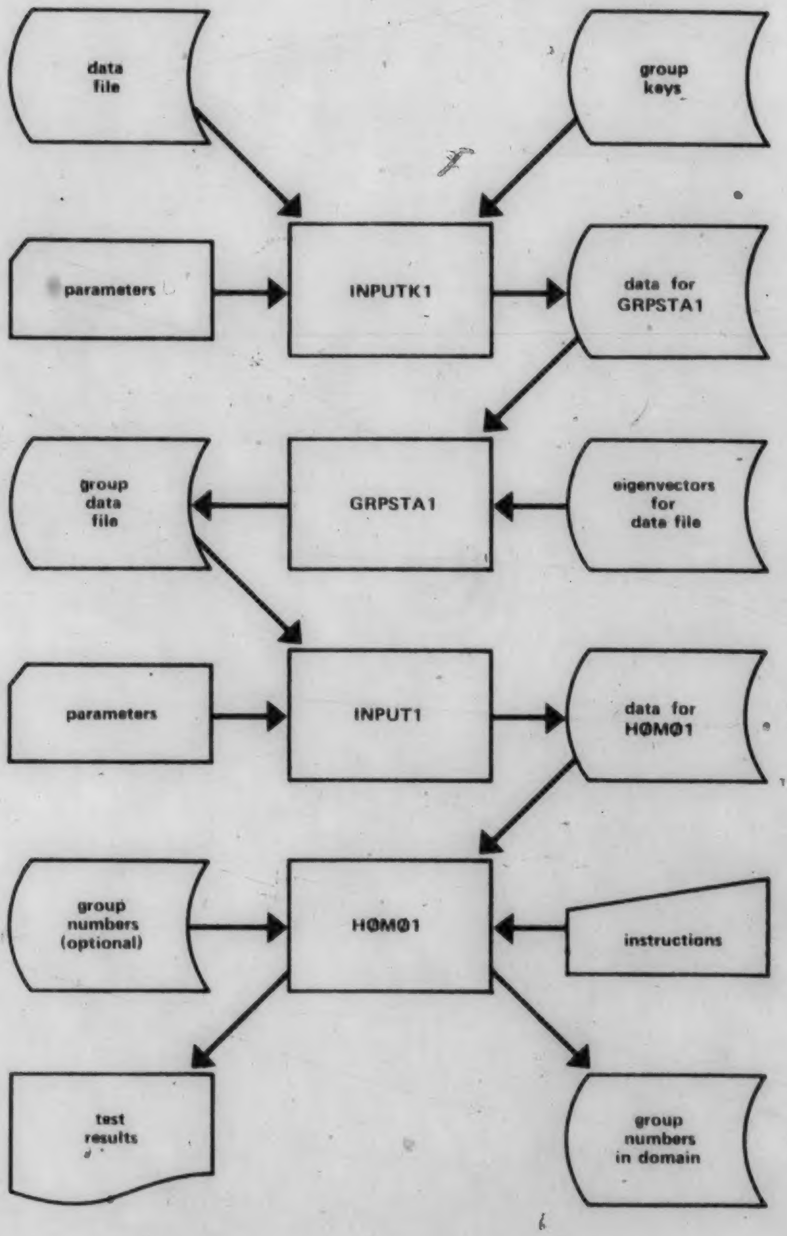

data file
group keys
parameters
INPUTK1
data for GRPSTA1
group data file
GRPSTA1
eigenvectors for data file
parameters
INPUT1
data for HØMØ1
group numbers (optional)
HØMØ1
instructions
test results
group numbers in domain

Figure 31. Procedure for Areal Analysis - Part 3

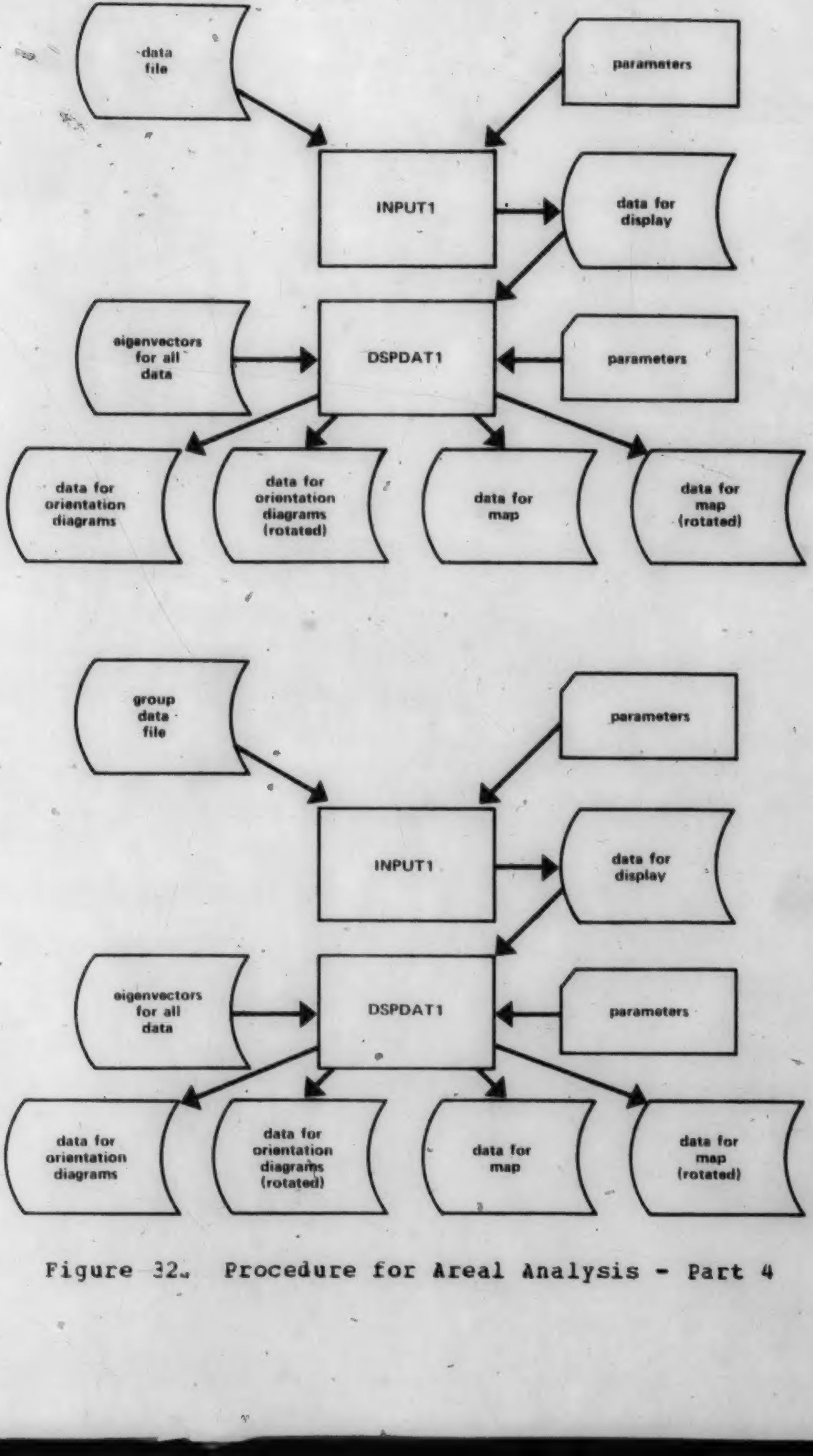

data file
parameters
INPUT1
data for display
eigenvectors for all data
DSPDAT1
parameters
data for orientation diagrams
data for orientation diagrams (rotated)
data for map
data for map (rotated)
group data file
parameters
INPUT1
data for display
eigenvectors for all data
DSPDAT1
parameters
data for orientation diagrams
data for orientation diagrams (rotated)
data for map
data for map (rotated)

Figure 32. Procedure for Areal Analysis - Part 4

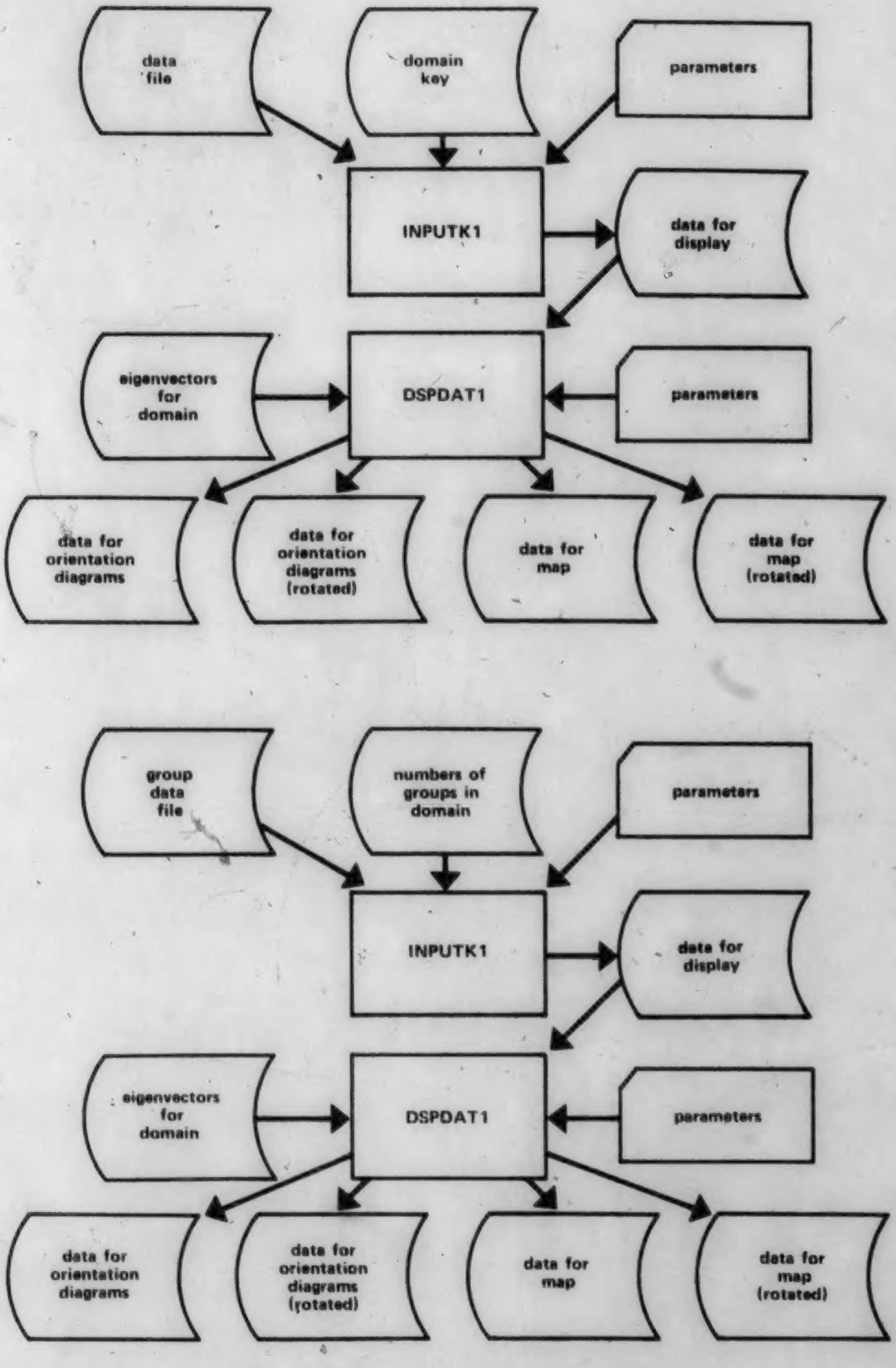

data file
domain key
parameters
INPUTK1
data for display
eigenvectors for domain
DSPDAT1
parameters
data for orientation diagrams
data for orientation diagrams (rotated)
data for map
data for map (rotated)
group data file
numbers of groups in domain
parameters
INPUTK1
data for display
eigenvectors for domain
DSPDAT1
parameters
data for orientation diagrams
data for orientation diagrams (rotated)
data for map
data for map (rotated)

Figure 33. Procedure for Areal Analysis - Part 5

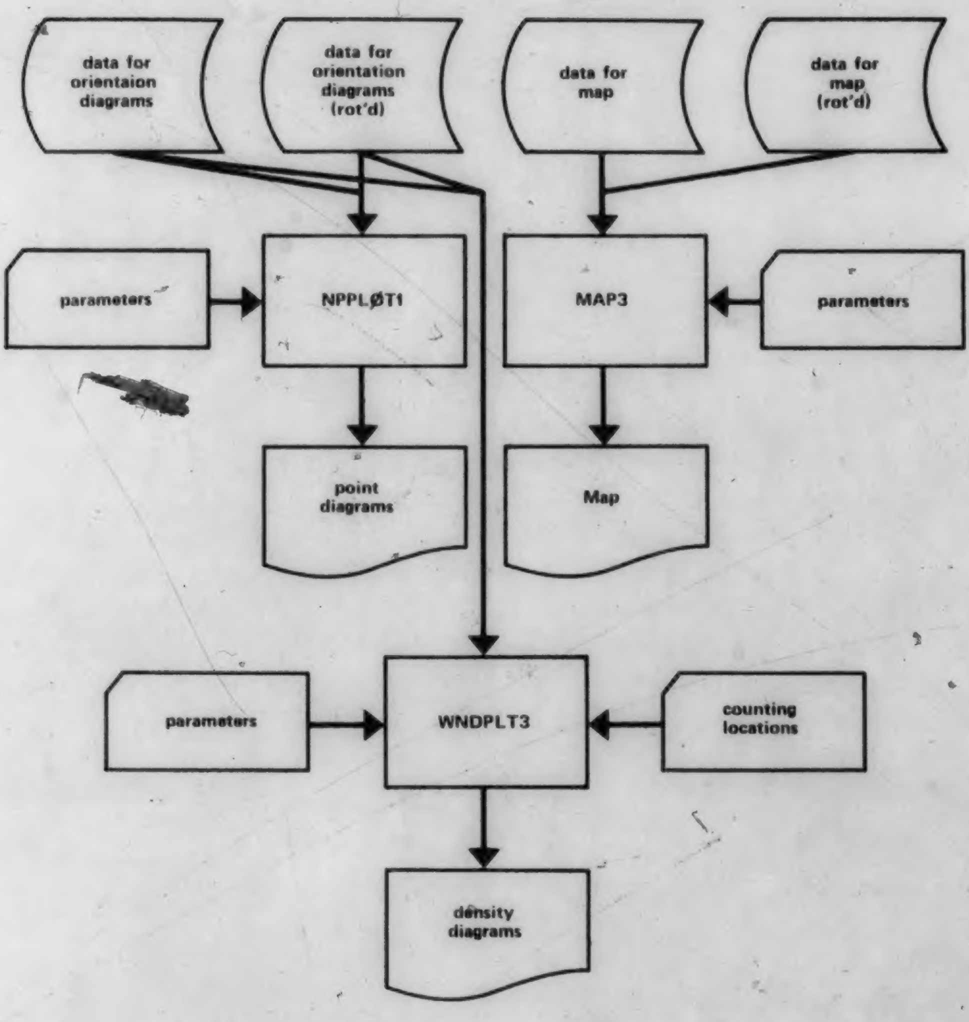

data for orientaion diagrams
data for orientation diagrams (rot'd)
data for map
data for map (rot'd)
parameters
NPPLØT1
MAP3
parameters
point diagrams
Map
parameters
WNDPLT3
counting locations
density diagrams

Figure 34. Procedure for Areal Analysis - Part 6

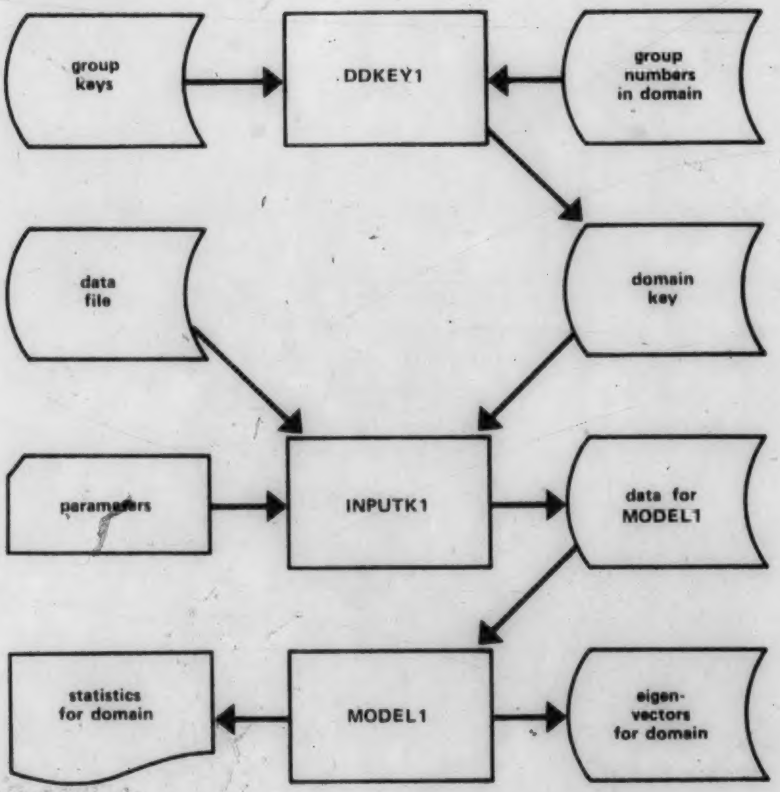

group keys
DDKEY1
group numbers in domain
data file
domain key
parameters
INPUTK1
data for MODEL1
statistics for domain
MODEL1
eigen-vectors for domain

Figure 35. Procedure for Areal Analysis - Part 7

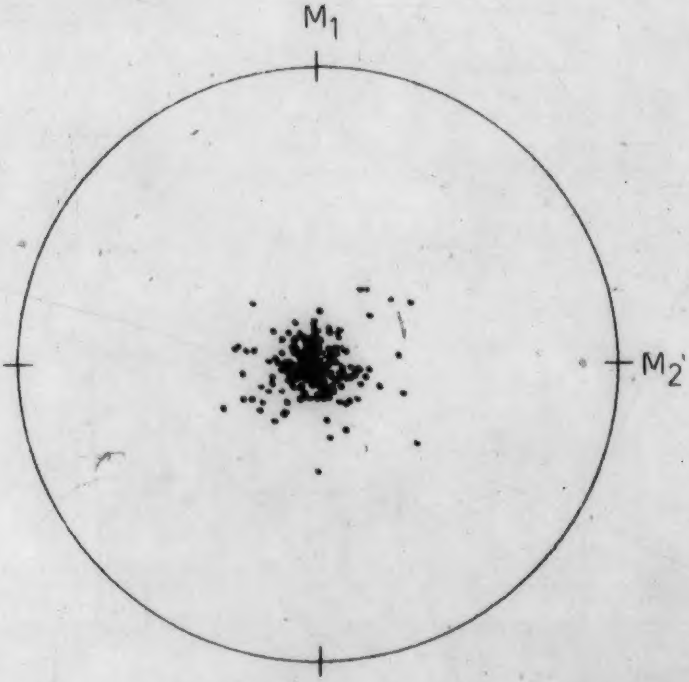


Figure 36. Lower-Hemisphere Equal-Area Projection of 297 Poles to Bedding in Domain 3 at the Fording Coal Mine after Rotation to Make the Mean Vertical

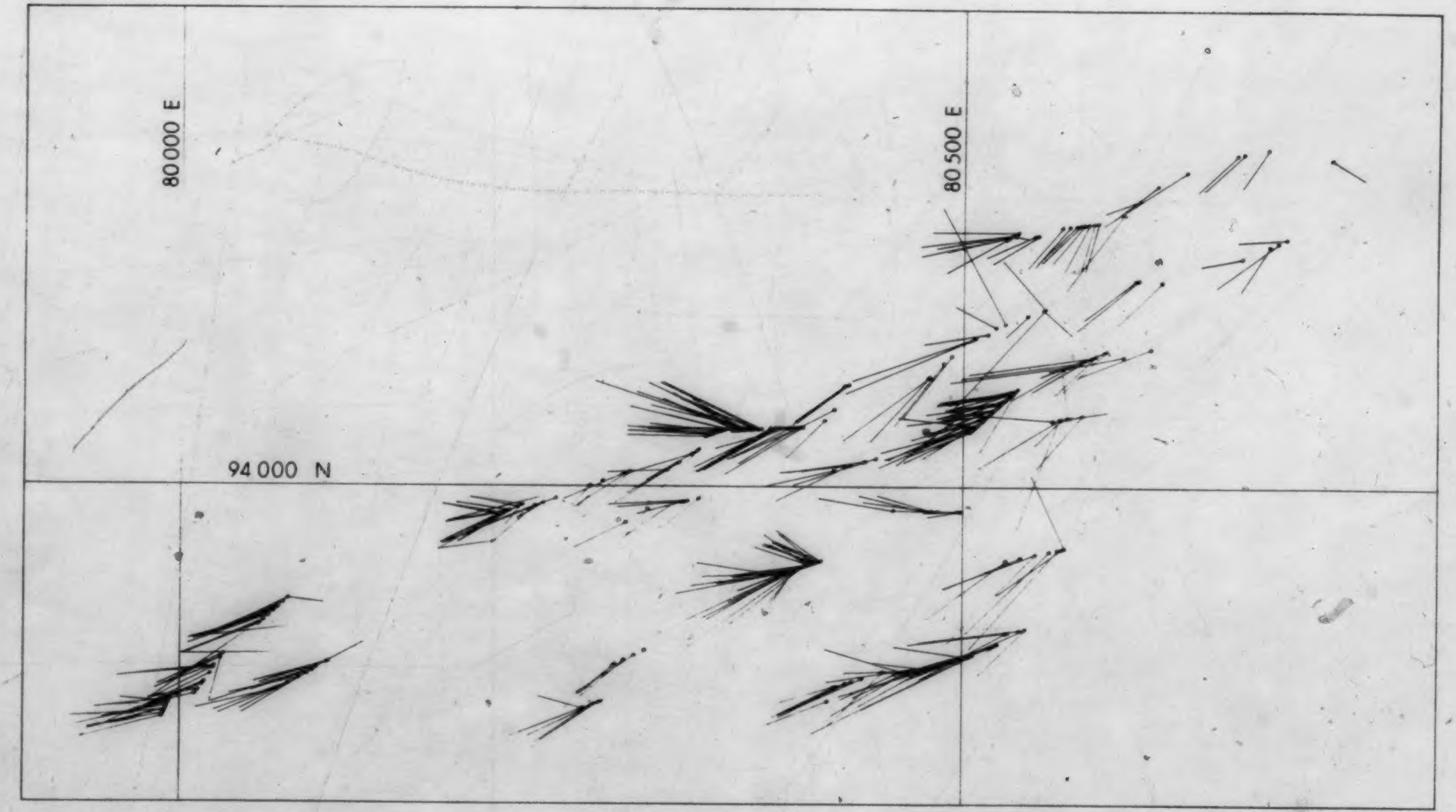

80000 E
80500 E
94000 N

Figure 37. Fording Bedding Data - Locations and Orientations of Observed Discontinuities

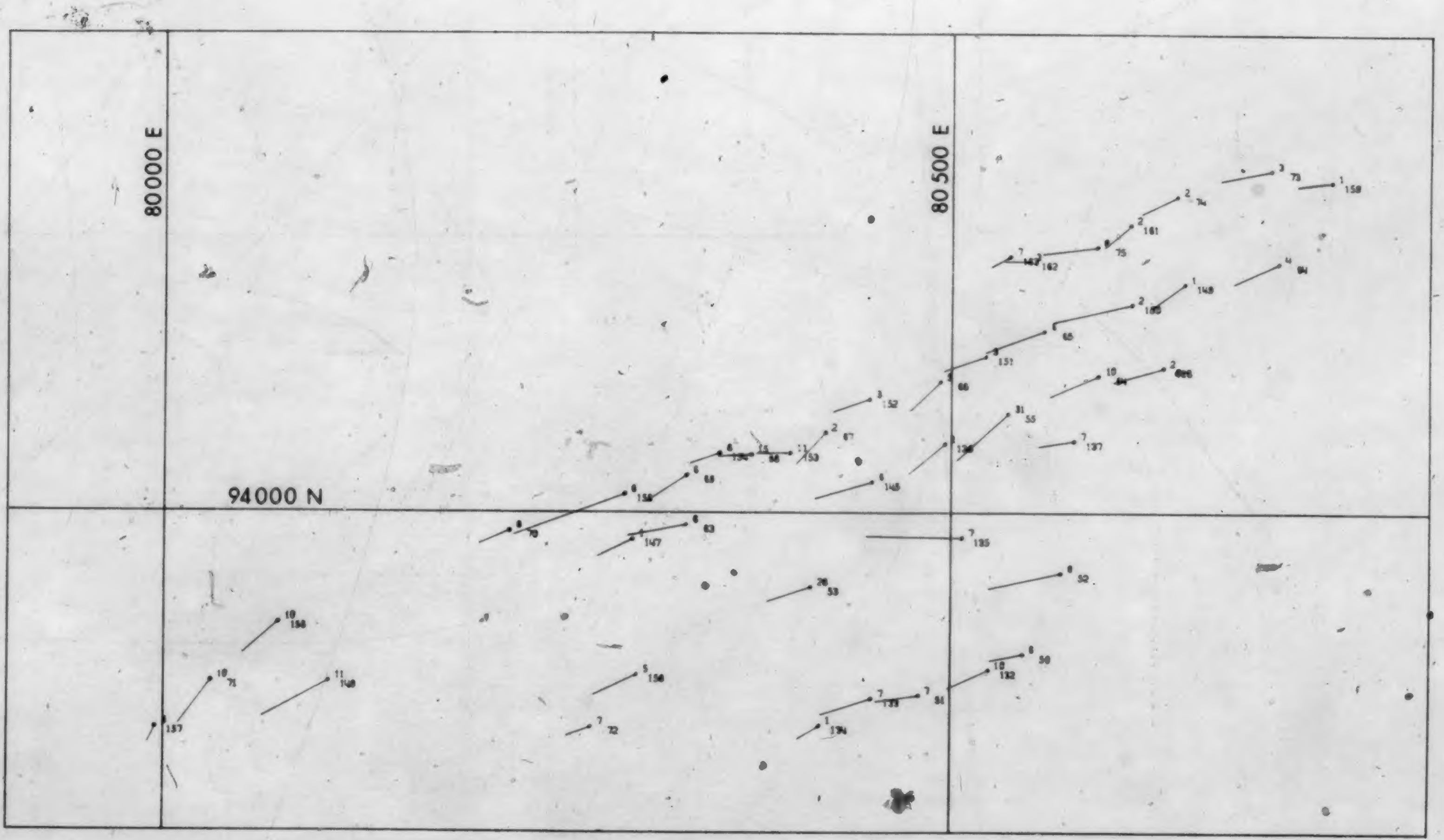

80000 E
80 500 E
94000 N

Figure 38. Fording Bedding Data - Traverses with Numbers of Observations (above) and Traverse Numbers (below)

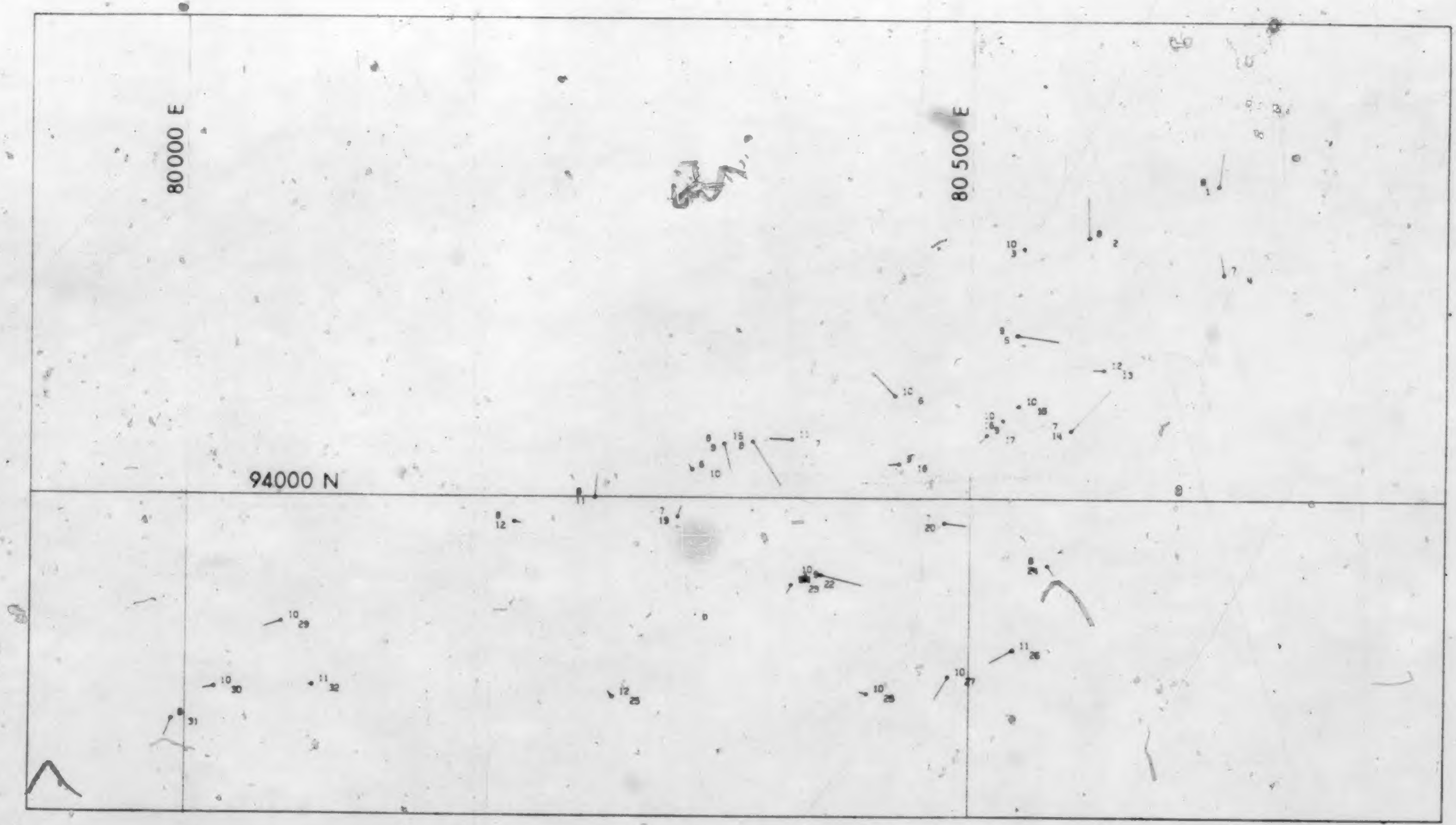

80000 E
80 500 E
94000 N

Figure 39. Fording Bedding Data - Deviations of Domain Means from Overall Mean

References Cited

Agterberg FP and Briggs G. 1963.
Statistical analysis of ripple marks in Atokan and Desmoinesian rocks in the Arkoma Basin of east-central Oklahoma. J Sed Petrol, 33, 393-410.

Agterberg FP, Hills LV and Trettin HP. 1967.
Paleocurrent trend analysis of a delta in the Bjorne Formation (Lower Triassic) of northeastern Melville Island, Arctic Archipelago. J Sed Petrol, 37, 852-862.

Anderson TW and Stephens MA. 1972.
Tests for randomness of directions against equatorial and bimodal alternatives. Biometrika, 59, 613-621.

Andrews JT. 1971.
Techniques of till fabric analysis. Brit Geomorph Res Grp Tech Bull 6, 43p.

Andrews JT and King CAM. 1968.
Comparative till fabrics and till fabric variability in a till sheet and a drumlin: a small-scale study. Proc Yorks Geol Soc, 36, 435-461.

Andrews JT and Shimizu K. 1966.
Three-dimensional vector technique for analyzing till fabrics: discussion and FORTRAN program. Geog Bull, 8, 151-165.

Andrews JT and Smith DI. 1966.
The variability of till fabric. Brit Geomorph Res Grp Occl Pap 3, 33-37.

Andrews JT and Smith DI. 1970.
Statistical analysis of till fabric: methodology, local and regional variability. Quart J Geol Soc Lond, 125, 503-542.

Baker DW, Wenk HR and Christie JM. 1969.
X-ray analysis of preferred orientation in fine-grained quartz aggregates. J Geol, 77, 144-172.

Barton CM. 1974.
Simplified procedures for the vector summation and statistical analysis of spherically distributed point clusters. Commonwealth Scientific and Industrial Res Organization, Australia, Div of Appl Geomechanics Tech Rept 20, 24p.

Bayrock LA and Hughes GM. 1962.
Surficial geology of the Edmonton district, Alberta. Res Coun Alberta Prelim Rept 62-6, 40p.

Berner H et al. 1972.
GEOMAP - A data system for geological mapping. Proc 24th Intl Geol Congr, Montreal, Sec 16, 3-11.

Billings MP. 1954.
Structural geology. Prentice-Hall Inc, New York (2nd ed).

Bingham C. 1964.
Distributions on the sphere and on the projective plane. Ph D diss, Yale U.

Box GEP and Jenkins GM. 1970.
Time series analysis, forecasting and control. Holden-Day, San Francisco, 553p.

Cacoullos T. 1966.
Estimation of a multivariate density. Ann Inst Statist Math, 18, 179-189.

Cowan WR. 1968.
Ribbed moraine: Till-fabric analysis and origin. Can J Earth Sci, 5, 1145-1159.

Cruden DM. 1968.
Methods of calculating the axes of cylindrical folds: a review. Geol Soc Amer Bull, 79, 143-148.

Cruden DM and Charlesworth HAK. 1966.
The Mississippian-Jurassic unconformity near Nordegg, Alberta. Bull Can Petrol Geol, 14, 266-272.

Cruden DM and Charlesworth HAK. 1972.
Observations on the numerical determination of axes of cylindrical folds. Geol Soc Amer Bull, 83, 2019-2024.

David HA. 1952.
Upper 5 and 1% points of the maximum F-ratio. Biometrika, 39, 422-424.

Dimroth E. 1962.
Untersuchungen zum Mechanismus von Blastesis und syntexis in Phylliten und Hornfelsen des sudwestlichen Fichtelgebirges I. Die statistische Auswertung einfacher Gurteldiagramme. Tscherm Min Petr Mitt, 8, 248-274.

Dimroth E. 1963.
Fortschritte der Gefugestatistik. N Jb Min Mh, 13, 186-192.

Elkins TA. 1968.
Cubical and spherical estimation of a multivariate probability density. J Amer Statist Assoc, 63, 1495-1513.

Elliot D. 1965.
The quantitative mapping of directional minor structures. J Geol, 73, 865-880.

Epanechnikov VA. 1969.
Nonparametric estimation of a multivariate probability density. Theory Prob Appl (USSR), 14, 153-158.

Fara HD and Scheidegger AE. 1963.
An eigenvalue method for the statistical evaluation of fault plane solutions of earthquakes. Seismol Soc Amer Bull, 53, 811-816.

Fisher RA. 1953.
Dispersion on a sphere. Proc Roy Soc Lond, A217, 295-306.

Flinn D. 1958.
On tests of significance of preferred orientation in three-dimensional fabric diagrams. J Geol, 66, 526-539.

Fox WT. 1967.
Fortran IV program for vector trend analyses of directional data. Kansas Geol Surv Computer Sci Contr 11.

Fraser DAS. 1958.
Statistics: An introduction. John Wiley & Sons, New York, 398pp.

Fukunaga K and Hostetler LD. 1973.
Optimization of k-nearest-neighbor density estimators. IEEE Trans on Info Theory, IT19, 320-326.

Gessaman MP. 1970.
A consistent non-parametric multivariate density estimator based on statistically equivalent blocks. Ann Math Statist, 41, 1344-1346.

Glenn JW, Donner JJ and West RG. 1957.
On the mechanism by which stones in till become oriented. Amer J Sci, 255, 194-205.

Harrison PW. 1957.
A clay-till fabric: Its character and origin. J Geol, 65, 275-308.

Hartley HO. 1950.
The maximum F-ratio as a short-cut test for heterogeneity of variance. Biometrika, 37, 308-312.

Haugh I, Brisbin WC and Turek A. 1967.
A computer-oriented field data sheet for structural data. Can J Earth Sci, 4, 657-662.

Holmes CD. 1941.
Till fabric. Geol Soc Amer Bull, 52, 1299-1354.

Kamb WB. 1959.
Ice petrofabric observations from Blue Glacier, Washington, in relation to theory and experiment. J Geophys Res, 64, 1891-1909.

Kauranne LK. 1960.
A statistical study of stone orientation in glacial till. Comm Geol Finland Bull, 188, 87-97.

LaFountain LJ. 1970.
Plotted and point-counted stereograms by computer X-Y plotter or microfilm devices. Geol Soc Amer Bull, 81, 1267-1272.

Lambert MB and Reesor JE. 1974.
A format for field and laboratory data collection. In Gordon T and Hutchison WW (eds), Computer use in projects of the Geological Survey of Canada. Geol Surv Can Pap 74-60.

Laurin AF et al. 1972.
Application of data processing techniques in the Grenville Province, Quebec, Canada. Proc 24th Intl Geol Congr, Montreal, Sec 16, 22-35.

Lindsay JF. 1970.
Clast fabric of till and its development. J Sed Petrol, 40, 629-641.

Loftsgaarden DO and Quesenberry CP. 1965.
A nonparametric estimate of a multivariate density function. Ann Math Statist, 36, 1049-1051.

Mardia KV. 1972.
Statistics of directional data. Academic Press, London and New York, 357p.

Mark DM. 1971.
Rotational vector procedure for the analysis of till fabrics. Geol Soc Amer Bull, 82, 2661-2666.

Mark DM. 1973.
Analysis of axial orientation data, including till fabrics. Geol Soc Amer Bull, 84, 1369-1374.

Mellis O. 1942.
Gefugediagramme in stereographischer projektion. Mineral Petrog Mitt, 53, 330.

Muecke GK and Charlesworth HAK. 1966.
Jointing in folded Cardium sandstones along the Bow River, Alberta. Can J Earth Sci, 3, 579-596.

Noble DC and Eberly SW. 1964.
A digital computer procedure for preparing beta diagrams. Amer J Sci, 262, 1124-1129.

Piteau DR. 1973.
Characterizing and extrapolating rock joint properties in engineering practice. Rock Mechanics Suppl 2, 5-31.

Piteau DR and Russel L. 1971.
Cumulative sums technique: A new approach to analyzing joints in rock. 13th Symp Rock Mechanics, Univ of Illinois.

Ramsden J. 1970.
Till fabric studies in the Edmonton area, Alberta, with special emphasis on methodology. M Sc thesis, Univ of Alberta, Edmonton.

Roberts MC and Mark DM. 1970.
Use of trend surfaces in till fabric analysis. Can J Earth Sci, 7, 1179-1184.

Robinson P, Robinson R and Garland SJ. 1963.
Preparation of beta diagrams in structural geology by a digital computer. Amer J Sci, 261, 913-928.

Roddick JA and Hutchison WW. 1972.
A computer-based system for geological field data on the Coast Mountains Project, British Columbia, Canada. Proc 24th Intl Geol Congr, Montreal, Sec 16, 36-46.

Scheidegger AE. 1964.
The tectonic stress and tectonic motion direction in Europe and western Asia as calculated from earthquake fault plane solutions. Seismol Soc Amer Bull, 54, 1519-1528.

Scheidegger AE. 1965.
On the statistics of the orientation of bedding planes, grain axes, and similar sedimentological data. US Geol Surv Prof Pap 525-C, 164-167.

Selby B. 1964.
Girdle distributions on a sphere. Biometrika, 51, 381-392.

Spencer AB and Clabaugh PS. 1967.
Computer program for fabric diagrams. Amer J Sci, 265, 166-172.

Starkey J. 1969.
A computer programme to prepare orientation diagrams. In Paulitsch P (ed), Experimental and natural rock deformation. Springer, Verlag, 51-74.

Starkey J. 1975.
The quantitative analysis of orientation patterns represented in equal-area, spherical projection. Geol Soc Amer Abstracts with Programs, 7, 863 (Abstr).

Stauffer MR. 1966.
An empirical-statistical study of three-dimensional fabric diagrams as used in structural analysis. Can J Earth Sci, 3, 473-498.

Stephens MA. 1965.
The goodness-of-fit statistic V_n: distribution and significance points. Biometrika, 52, 309-321.

Stephens MA. 1967.
Tests for the dispersion and for the modal vector of a distribution on a sphere. Biometrika, 54, 211-223.

Stephens MA. 1969.
Multi-sample tests for the Fisher distribution for directions. Biometrika, 56, 169-181.

Stephens MA. 1970.
Use of the Kolmogorov-Smirnov, Cramer-von Mises and related statistics without extensive tables. J Roy Statist Soc, B32, 115-122.

Strand T. 1945.
A method of counting out petrofabric diagrams. Nordsk Geol Tidskr, 24, 112-114.

Sturm E. 1971.
High-resolution paleocurrent analysis by moving vector averages. J Geol, 79, 222-233.

Terzaghi RD. 1965.
Sources of error in joint surveys. Geotechnique, 15, 287-304.

Thompson PD. 1956.
Optimum smoothing of two-dimensional fields. Tellus, 8, 384-393.

Turner FJ and Weiss LE. 1963.
Structural analysis of metamorphic tectonites. McGraw-Hill, New York.

Warner J. 1969.
FORTRAN IV program for construction of pi diagrams with the Univac 1108 computer. Kansas Geol Surv Computer Contr 33, 38p.

Watson GS. 1956a.
A test for randomness of directions. Roy Astron Soc Monthly Notices Geophys Suppl, 7, 160.

Watson GS. 1956b.
Analysis of dispersion on a sphere. Roy Astron Soc Monthly Notices Geophys Suppl, 7, 153-159.

Watson GS. 1960.
Further significance tests on a sphere. Biometrika, 47, 87-91.

Watson GS. 1965.
Equatorial distributions on a sphere. Biometrika, 52, 193-203.

Watson GS. 1966.
The statistics of orientation data. J Geol, 74, 786-797.

Watson GS. 1970.
Orientation statistics in the earth sciences. Univ Upsala Geol Instn Bull NS2, 9, 73-89.

Watson GS and Irving E. 1957.
Statistical methods in rock magnetism. Roy Astron Soc Monthly Notices Geophys Suppl, 7, 289-300.

Watson GS and Leadbetter MR. 1962.
On the estimation of the probability density. I. Ann Math Statist, 34, 480-491.

Watson GS and Williams EJ. 1956.
On the construction of significance tests on the circle and the sphere. Biometrika, 43, 344-352.

Wegman EJ. 1972.
Nonparametric probability density estimation: I. A summary of available methods. Technometrics, 14, 533-546.

Whittle P. 1958.
On the smoothing of probability density functions. J Roy Statist Soc, B20, 334-343.

Glossary of Symbols

A symbol is followed by a brief indication of its meaning and then by the number of the page where it is defined or first used in the text. Except for the standard symbols, lower case Roman letters denote scalars, upper case Roman letters denote vectors or matrices, and Greek letters denote angles.

Note: A' is the transpose of A. $\hat{k}$ is an estimate of k.

a	area of counting circle. 23.
a_o	the area of the counting circle corresponding to a value v_o of v. 36.
b	a measure of the bias of $\hat{d}$ as an estimator of d. 31.
c	constant defined in context.
d	population density at a point. 23.
d_f	the peak density of a unimodal Fisher-distributed population after smoothing by the Fisher-weighted density estimator. 34.
d_f^*	the peak density for a single Fisher-distributed point cluster in a multimodal population after smoothing by the Fisher-weighted density estimator. 35.
d_s	peak density of a unimodal Fisher-distributed population after smoothing by the constant-area density estimator. 32.
d_s^*	the peak density for a single Fisher-distributed point cluster in a multimodal population after smoothing by the constant-area density estimator. 33.
d_v	the peak density of a unimodal Fisher-distributed population after smoothing by the constant-proportion density estimator. 35.
d_v^*	the peak density for a single Fisher-distributed point cluster in a multimodal population after smoothing by the constant-proportion density estimator. 36.
g	defined in context. 92.
h	defined in context. 92.
k	concentration parameter of a probability

	distribution. 27.
k_p	concentration parameter of a Fisher-distributed point cluster. 32.
k_w	concentration parameter of the Fisher density function used as a weighting function in the Fisher-weighted density estimator. 34.
k_1, k_2, k_3	the concentration parameters of Bingham's distribution. 65.
l	Kuiper's test statistic. 71.
l_{max}, l_{min}	maximum and minimum differences between theoretical and empirical cumulative distributions. 71.
m	the length of M. 62.
m_1, m_2, m_3	the eigenvalues of T in increasing order. 63.
n	number of points in sample. 23.
p	proportion of population inside the counting circle. 24.
p_c	proportion of population point cluster 1 inside the counting circle. 33.
p_o	the value of p corresponding to a value v_o of v. 29.
p_1	proportion of a population consisting of several Fisher-distributed point clusters in point cluster 1. 33.
q	number of samples. 92.
r	uniform random number on the interval (0,1). 36.
t	number of points inside counting circle. 23.
t_{max}	largest value of t for several counting locations. 25.
$t_1, \ldots t_{360}$	relative frequencies. 81.
u	distance from centre of projection to the projected point (θ, ϕ). 80.
u_o	radius of projection. 80.
v	coefficient of variation. 24.

v	required coefficient of variation of $\hat{d}$. 29.
w()	weighting function. 23.
$w_1 \ldots w_n$	weights associated with observed unit vectors. 75.
x,y,z	direction cosines. 36.
x_p, y_p	coordinates on a projection of the projected point (θ,ϕ). 80.
A	any unit vector. 62.
M	the mean vector of a sample of unit vectors. 62.
M_1, M_2, M_3	the eigenvectors of T in order of increasing eigenvalues. 63.
P	observed point. 27.
$P_1 \ldots P_n$	a sample of observed points. 28.
Q	true position of the observed point P. 27.
T	variance-covariance matrix of the components of a sample of unit vectors. 63.
U	a fixed unit vector. 63.
U,V,W	orthogonal fixed unit vectors. 65.
X,Y,Z	orthogonal unit vectors denoting fixed reference directions. 36
β	angle between traverse and planes. 14.
γ	angle between centre of population and counting location. 37.
ϕ	spherical coordinate. 36.
θ	angle between two vectors; spherical coordinate. 23; 36.
θ_c	radius of counting circle. 23.
$\theta_{\frac{1}{2}}$	the distance from the centre of a Fisher distribution at which the density is half the peak density. 47.

Standard Symbols

e	2.71828...
$\exp()$	power of e.
$F_{df1,df2}$	the F-distribution with df1 and df2 degrees of freedom.
F_{α}	the upper 100α% point of the F-distribution.
i	index.
j	index.
$\ln()$	natural logarithm.
$\Pr()$	probability of an event.
α	the upper tail probability of a probability distribution.
π	3.14159265...
Σ	summation.
χ^2_{df}	the chi-square distribution with df degrees of freedom.

Appendix 1

Documentation of Computer Programs

Program .... Page

CONVERT3 168

DATFIX1 200

DDKEY1 207

DSEM 212

DSPDAT1 221

GRPKEY1 228

GRPSTA1 235

HOMO 241

INPUTK1 257

INPUT1 267

KEY4 275

MAP3 318

MODEL1 331

NPPLOT1 337

ROSE 345

SAMPLEGEN 354

TRVLST2 361

WNDPICT3 367

SECTION 1: PROGRAM IDENTIFICATION

PROGRAM TITLE:

Building of Structural Data File from Field Data

PROGRAM CODE NAME:

CONVERT3

WRITER:

John Ramsden

ORGANIZATION:

Department of Geology, University of Alberta, Edmonton, Alberta

DATE:

11 January 1974

VERSION:

Zero

SOURCE LANGUAGE:

FORTRAN IV

AVAILABILITY:

Available from Dr. G. Herget
Elliot Lake Laboratory
CANMET, EMR
P.O. Box 100
Elliot Lake, Ontario P5A 2J6

ABSTRACT:

The program converts field data on discontinuities in a rock mass collected on straight line traverses to a form suitable for subsequent processing.

The data is collected as described in the Discodat (1973) manual. For each traverse, the program merges the information recorded on the type 1 form with that recorded on the type 2 forms to produce a set of identical-format data records, one for each observed discontinuity.

SECTION 2: ENGINEERING DOCUMENTATION

NARRATIVE DESCRIPTION:

The Traverse: The information from the type 1 input record is placed in the output record unchanged, with the following exceptions. The grid co-ordinates and elevation of the traverse origin are converted to the basic unit of the survey using the scale factors read from the parameter card. They are not placed in the output record. The reference direction is not placed in the output record.

The Observations:

(a) Sequence Number

The observations in a traverse are numbered sequentially in the order of input. This number is placed in the output record.

(b) Distance

The distance of the point of observation from the traverse origin is converted to the basic unit of the survey using the traverse scale factor. The converted distance is placed in the output record.

(c) Grid Co-ordinates and Elevation

The program computes the grid co-ordinates and elevation of each point of observation as follows. The distance from the traverse origin to the point of observation is resolved into northward, eastward, and upward components, which are added to the co-ordinates and elevation of the traverse origin. The three co-ordinates are rounded to the nearest integer for output.

(d) Orientation

The program computes the dip direction of the discontinuity by adding the recorded azimuth to the reference direction. Direction cosines of the downward-directed normal to the discontinuity are computed, multiplied by 10000 and rounded to the nearest integer for output. An upper limit of 9999 is imposed.

(e) Terzaghi Correction Factor (Weight)

The Terzaghi correction factor compensates for bias due to the orientation of the traverse. Its value is the reciprocal of the cosine of the angle between the traverse and the normal to the discontinuity. It is truncated at an angle of 85 degrees, and in order to provide integer output is multiplied by ten and rounded to the nearest integer. The number placed in the output record thus has a range of from 10 to 115. This factor may be used as a weighting factor in subsequent processing.

(f) Linears

Direction cosines of a linear feature are computed from the dip direction and dip of the discontinuity and the pitch of the linear as follows. If the strike vector S is rotated counterclockwise about the upward-directed normal to the discontinuity through an angle equal to the pitch of the line L, it will then coincide with that line. The matrix RM that will perform this rotation is obtained by calling the subroutine ROTMAX. The direction cosines of the line are then found by the matrix multiplication,

$$RM \times S \rightarrow L$$

which is performed by the SSP subroutine GMPRD.

METHOD OF SOLUTION:

Borehole Data: The orientations of discontinuities observed in drill cores are computed as follows. Mutually perpendicular axes U, V, W are defined with respect to the core, so that U, V, W are a righthanded set; W is parallel to the core axis and is positive in the direction of drilling; U lies in the vertical plane containing W and is positive upward. If W is vertical, the direction of U is arbitrary.

The orientations of the discontinuities in the core are measured with the core oriented so that U, V, W are parallel to the standard reference axes: north, east, down, respectively. To obtain the original orientations of the discontinuities, we must apply to the recorded orientations the rotation that will bring the standard reference axes into coincidence with the original orientations of U, V, W. To be able to perform this rotation it is necessary only to know the orientation of the borehole and the direction of drilling.

If the trend and plunge of the boreholes are T, P then the original directions of U, V, W are

	U	V	W
Trend	T	T+90°	T
Plunge	P-90°	0	P
Direction Cosines			
1	cosT sinP	-sinT	cosT cosP
2	sinT sinP	cosT	sinT cosP
3	-cosP	0	sinP

If a 3 x 3 identity matrix, whose columns are the reference axes, is pre-multiplied by M, the direction cosine matrix in the above table whose columns are U, V, W the result will be M. Thus, pre-multiplication of a column vector by M performs the required rotation. For each discontinuity, the program calls the SSP subroutine GMPRD to pre-multiply by M the column vector consisting of the direction cosine of the normal to the discontinuity in its recorded orientation. The resulting column vector contains the direction cosines of the normal in its original orientation. From these are computed the original dip direction and dip.

PROGRAM CAPABILITIES:

Data Restrictions: Not more than 99 observations may be processed as one traverse. There is no limit to the number of traverses that may be processed in one run. All other data restrictions are as described in the Discodat (1973) manual.

DATA INPUTS: The input data are fully described in the Discodat (1973) manual, except for the case of borehole data which should be recorded as follows.

A reference direction of 999 indicates borehole data. To record borehole data, draw a reference line along the side of the core, on that part that was the top of the core in its original orientation. Now orient the core so that its axis is vertical (drilling direction downward) and the reference line is on the north side. Measure the dip-direction and dip of each discontinuity in this orientation, recording these in the 'azimuth' and 'dip' fields. Record the trend and plunge of the borehole in the 'traverse-trend' and 'traverse-plunge' fields.

If the drilling direction was vertically downward, draw the reference line on the original north side of the core, and enter zero in the 'traverse trend' field and 90 in the 'traverse plunge' field. If the direction was vertically upward, draw the reference line on the original south side of the core and enter zero in the 'traverse trend' field and -90 in the 'traverse plunge' field.

Input data records are of two types. Each type 1 record contains information pertaining to one traverse, and is followed immediately by the type 2 records for that traverse; each type 2 record contains information pertaining to one observation. The format of the type 1 record is as follows. (A) indicates alphanumeric data.

Columns	Contents
7-13	Traverse Identification (A)
15-19	Elevation
21-22	Domain (A)
24-26	Formation (A)
28-31	Pit Bench Level (A)
33-35	Pit Bench Location (A)
37-42	Local Grid Easting
44-49	Local Grid Northing
51-53	Traverse Trend
55-57	Traverse Plunge
59-62	Traverse Length
64	Traverse Scale
66-67	Number of Observations
69-71	Reference Direction

Special conditions: A reference direction of 999 indicates borehole data.

The format of the type 2 record is as follows.

Columns	Contents
7-10	Distance
12-13	Type (A)
15-17	Azimuth
19-21	Dip
23	Size A
25	Spacing (A)
27	Filling 1 (A)
29	Filling 2 (A)
31	Water (A)
33-35	Waves
37-38	Line Type (A)
40-42	Line Pitch
44-46	Lithology (A)
48-49	Hardness (A)

Special conditions: If 'distance' is blank, zero or negative the distance is taken as zero. If any one of the co-ordinates and elevation fields on the type 1 input record is blank or zero, the co-ordinates of all observations on the traverse are set equal to those of the traverse origin.

In addition to the above data cards, one parameter card is required containing scale factors for the grid co-ordinates and elevation, in the following format.

Columns	Contents
1	Scale Factor for Grid Co-ordinates
2	Scale Factor for Elevation

These scale factors are determined in the same way as the traverse scale factors, as described on page A8 of the 1973 Discodat manual.

Program Options: None.

Printed Output: Each type 1 input record is printed as it is read.

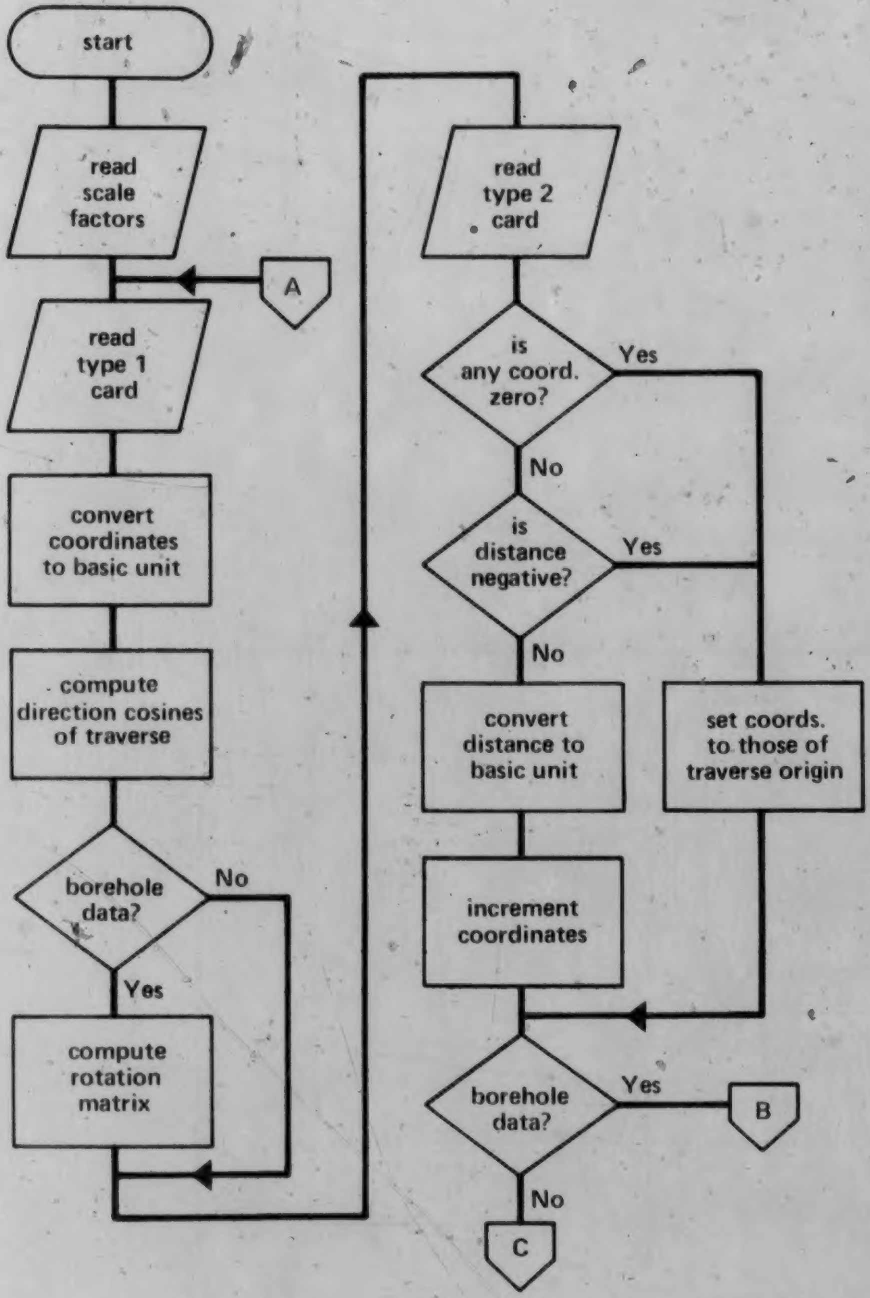
Other Outputs: The converted data records, one for each observation, are normally written on magnetic tape. The record format is as follows.

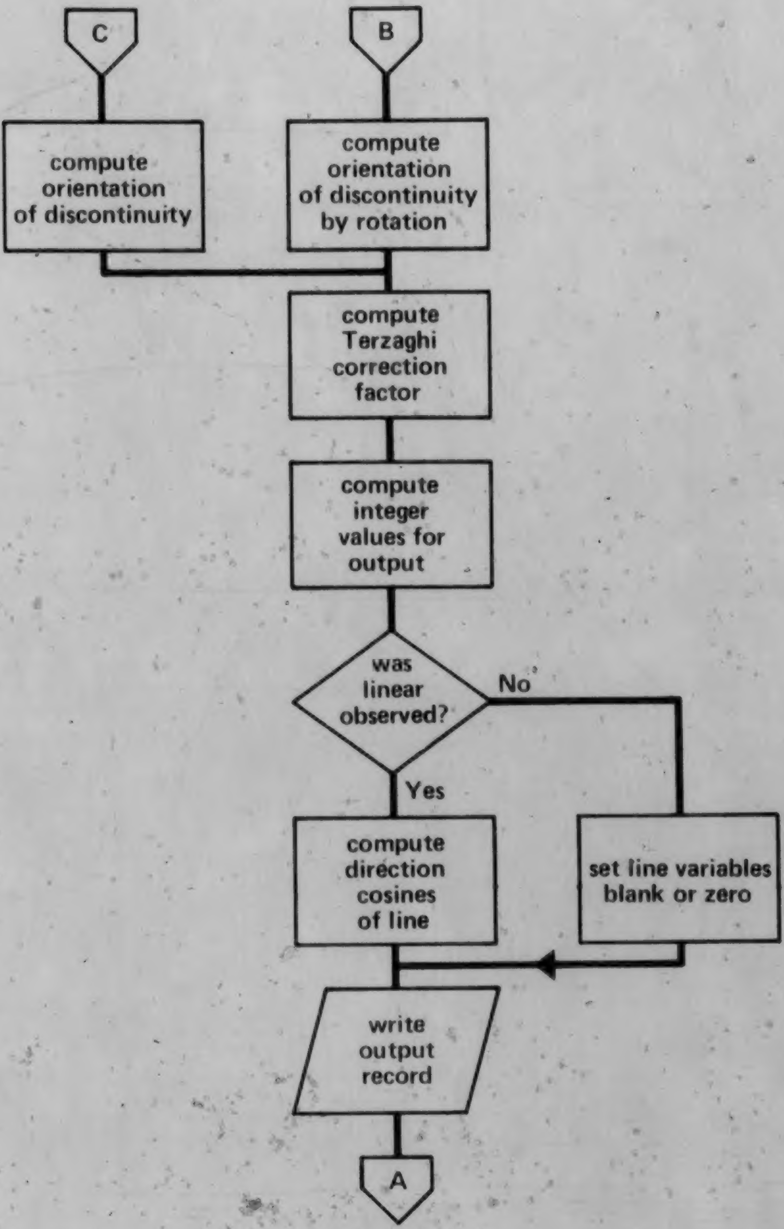
Columns	Contents
1	'D'
2- 8	Traverse Identification
9-10	Domain
11-13	Formation
14-17	Pit Bench Level
18-20	Pit Bench Location
21-23	Traverse Trend
24-26	Traverse Plunge
27-30	Traverse Length
31-32	Number of Observations
33	Traverse Scale
34-35	Sequence Number of Observation within Traverse
36-39	Distance

Columns	Contents
40-45	Easting
46-51	Northing
52-56	Elevation
57-58	Type
59-61	Dip Direction
62-64	Dip
65-69	a
70-74	b
75-79	g
80-82	Weight
83	Size
84	Spacing
85	Filling 1
86	Filling 2
87	Water
88-90	Waves
91-92	Linetype
93-97	l
98-102	m
103-107	n
108-110	Lithology
111-112	Hardness

a, b, g are the direction cosines of the downward-directed normal to the discontinuity. l, m, n are the direction cosines of the linear. If no linear was observed, l, m, and n are set to zero.

FLOW CHART


start
read scale factors
A
read type 1 card
convert coordinates to basic unit
compute direction cosines of traverse
borehole data?
No
Yes
compute rotation matrix
read type 2 card
is any coord. zero?
Yes
No
is distance negative?
Yes
No
convert distance to basic unit
set coords. to those of traverse origin
increment coordinates
borehole data?
Yes
B
No
C


C
B
compute orientation of discontinuity
compute orientation of discontinuity by rotation
compute Terzaghi correction factor
compute integer values for output
was linear observed?
No
Yes
compute direction cosines of line
set line variables blank or zero
write output record
A

SAMPLE RUN: Parameter Card

10

Data

```
*0101/2DM0002/59421/01/084/5920/   / 78383/ 96432/222/-02/1001/ /68/312/
*0201/  24/JN/130/+60/J/ / / /3/171/FD/160/SDS/R5/                    2DM0002
*0202/  28/JN/040/+87/J/ / / /3/164/FD/155/SDS/R5/                    2DM0002
*0203/  42/JN/055/-20/K/ / / /1/   /  /   /SDS/R5/                    2DM0002
*0204/  74/JN/143/+80/J/ / / /3/163/FD/ 90/SDS/R5/                    2DM0002
*0205/ 105/SP/145/+90/J/ / / /2/172/FD/165/SDS/R5/                    2DM0002
*0206/ 105/SP/145/+90/J/ / / /2/   /GV/ 93/SDS/R5/                    2DM0002
*0207/ 115/SP/150/+70/J/ / / /3/171/FD/160/SDS/R5/                    2DM0002
*0208/ 115/SP/150/+70/J/ / / /3/   /GV/ 80/SDS/R5/                    2DM0002
*0209/ 123/JN/153/+85/J/ / / /3/   /  /   /SDS/R5/                    2DM0002
*0210/ 155/JN/150/+85/J/ / / /3/172/FD/ 70/SDS/R5/                    2DM0002
*0211/ 180/JN/175/+80/J/ /F/ /3/174/FD/160/SDS/R5/                    2DM0002
*0212/ 181/SP/165/+68/J/ / / /3/   /GV/ 85/SDS/R5/                    2DM0002
*0213/ 223/SP/120/+60/K/ /M/B/4/173/FD/160/SDS/R5/                    2DM0002
*0214/ 223/SP/120/+60/K/ /M/B/4/   /GV/ 95/SDS/R5/                    2DM0002
*0215/ 228/JN/110/-20/J/ /B/M/4/   /  /   /SDS/R5/                    2DM0002
*0216/ 300/JN/ 90/-20/K/ / / /3/163/FD/180/SDS/R5/                    2DM0002
*0217/ 328/SP/130/+85/J/ /M/ /4/171/FD/180/SDS/R5/                    2DM0002
*0218/ 328/SP/130/+85/J/ /M/ /4/   /GV/ 90/SDS/R5/                    2DM0002
*0219/ 333/SP/140/+77/J/ / / /3/172/FD/170/SDS/R5/                    2DM0002
*0220/ 333/SP/140/+77/I/ / / /3/   /GV/ 85/SDS/R5/                    2DM0002
*0221/ 340/SP/145/+80/J/ / / /3/166/FD/ 90/SDS/R5/                    2DM0002
*0222/ 340/SP/145/+80/J/ / / /3/   /GV/ 90/SDS/R5/                    2DM0002
*0223/ 387/JN/150/+85/K/ / / /3/   /  /   /SDS/R5/                    2DM0002
*0224/ 410/SP/ 12/-73/K/ /F/ /3/   /GV/ 80/SDS/R5/                    2DM0002
*0225/ 416/JN/139/+69/K/ / / /3/170/FD/  5/SDS/R5/                    2DM0002
*0226/ 417/JN/152/ 82/J/ /F/ /3/166/FD/175/SDS/R5/                    2DM0002
*0227/ 434/JN/150/ 74/J/ /F/ /3/162/FD/ 84/SDS/R5/                    2DM0002
*0228/ 488/JN/ 82/-14/L/ /M/ /4/169/FD/155/SDS/R5/                    2DM0002
*0229/ 520/CN/ 95/-15/K/ / / /4/166/FD/180/SDS/R5/                    2DM0002
*0230/ 522/CN/ 95/-15/K/ / / /4/166/FD/180/SDS/S3/                    2DM0002
*0231/ 613/JN/ 24/-86/J/ / / /3/   /  /   /SDS/R5/                    2DM0002
*0232/ 597/JN/127/ 58/J/ /M/B/4/178/FD/  4/SDS/R5/                    2DM0002
*0233/ 585/JN/155/ 76/K/ / / /3/166/FD/  9/SDS/R5/                    2DM0002
*0234/ 614/JN/155/ 60/K/ / / /3/164/FD/  2/SDS/R5/                    2DM0002
*0235/ 647/JN/123/-18/L/ /B/ /4/   /  /   /   /  /                    2DM0002
*0236/ 660/JN/142/ 60/K/ / / /4/167/FD/180/SDS/R5/                    2DM0002
*0237/ 669/JN/ 32/-85/K/ / / /3/160/FD/170/SDS/R5/                    2DM0002
*0238/ 714/JN/149/ 87/K/ / / /3/171/FD/175/SDS/R5/                    2DM0002
*0239/ 747/JN/ 10/-75/K/ / / /3/177/FD/153/SDS/R5/                    2DM0002
*0240/ 760/CN/150/ 74/K/ / / /4/   /  /   /SDS/R5/                    2DM0002
*0241/ 768/CN/150/ 74/K/ / / /4/   /  /   /GVL/S1/                    2DM0002
*0242/ 798/SP/143/ 67/K/ / / /3/171/FD/174/SDS/R5/                    2DM0002
*0243/ 798/SP/143/ 67/K/ / / /3/   /GV/ 84/SDS/R5/                    2DM0002
*0244/ 824/SP/146/ 81/K/ / / /3/175/FD/175/SDS/R5/                    2DM0002
*0245/ 824/SP/146/ 81/K/ / / /3/   /GV/ 65/SDS/R5/                    2DM0002
*0246/ 846/SP/147/-77/J/ /B/M/2/171/FD/190/SDS/R5/                    2DM0002
*0247/ 846/SP/147/-77/J/ /B/M/2/   /GV/ 75/SDS/R5/                    2DM0002
*0248/ 856/JN/136/ 68/J/ / / /3/173/FD/180/SDS/R5/                    2DM0002
*0249/ 867/JN/ 18/-80/K/ / / /3/159/FD/112/SDS/R5/                    2DM0002
*0250/ 877/CN/137/ 55/J/ / / /3/162/FD/180/SDS/R5/                    2DM0002
*0251/ 878/CN/137/ 55/J/ / / /3/   /  /   /GVL/S1/                    2DM0002
*0252/ 902/JN/140/ 80/J/ / / /3/   /  /   /SDS/R5/                    2DM0002
*0253/ 929/JN/ 10/ 70/J/ / / /2/164/FD/165/SDS/R5/                    2DM0002
*0254/ 930/JN/ 25/-85/J/ / / /3/175/FD/165/SDS/R5/                    2DM0002
*0255/ 952/JN/150/[illegible]/J/ / / /3/   /  /   /SDS/R5/            2DM0002
*0256/ 962/JN/130/ 75/J/ / / /3/   /  /   /SDS/R5/                    2DM0002
*0257/ 968/JN/125/ 75/I/ / / /3/   /  /   /SDS/R5/                    2DM0002
```

```
*0258/ 980/JN/140/ 85/K/ / / /3/   /  /   /SDS/R5/                   2DM0002
*0259/ 981/JN/140/-15/K/ / / /2/   /  /   /SDS/R5/                   2DM0002
*0260/ 982/JN/ 75/-85/J/ / / /3/170/FD/160/SDS/R5/                   2DM0002
*0261/ 983/JN/140/ 90/J/ / / /3/163/FD/180/SDS/R5/                   2DM0002
*0262/ 995/JN/135/ 90/J/ / / /3/   /  /   /SDS/R5/                   2DM0002
*0263/1000/JN/ 65/-80/I/ / / /2/   /  /   /SDS/R5/                   2DM0002
*0264/1001/JN/115/-05/K/ / / /2/167/FD/180/SDS/R5/                   2DM0002
*0265/    /BG/105/-20/ /D/ / / /   /  /   /SDS/R5/                   2DM0002
*0266/    /BG/055/-20/ /D/ / / /   /  /   /SDS/R5/                   2DM0002
*0267/    /BG/115/-10/ /D/ / / /   /  /   /SDS/R5/                   2DM0002
*0268/    /BG/ 90/-20/ /D/ / / /   /  /   /SDS/R5/                   2DM0002
```

```
D2DM0002010B45920   222 -21001680 1  2478381496430259422JN 82 60-1205-8576 5000 15J    3171FD-9067 3301 2962SDSR5
D2DM0002010B45920   222 -21001680 2  2978381196429959422JN352 97-9999 1390  523 16J    3164FD 1480 8944 4220SDSR5
D2DM0002010B45920   222 -21001680 3  4278380296428959422JN187 20 3395  417 9397 32K    1         0    0    0SDSR5
D2DM0002010B45920   222 -21001680 4   7478378136426559424JN 95 80  658-9811 1736 17J    3163FD -151 1730 9848SDSR5
D2DM0002010B45920   222 -21001680 5 10578376036424259425SP 97 90 1219-9925    0 17J    2172FD-9587-1177 2588SDSR5
D2DM0002010B45920   222 -21001680 6 10578376036424259425SP 97 90 1219-9925    0 17J    2   GV 1210  149 9925SDSR5
D2DM0002010B45920   222 -21001680 7 11578375396423559425SP102 70 1954-9192 3420 22J    3171FD-9435 -809 3214SDSR5
D2DM0002010B45920   222 -21001680 8 11578375396423559425SP102 70 1954-9192 3420 22J    3   GV  993 3656 9254SDSR5
D2DM0002010B45920   222 -21001680 9 12378374896422959425JN105 85 2578-9623  872 22J    3         0    0    0SDSR5
D2DM0002010B45920   222 -2100168010 15578372696420559426JN102 85 2071-9744  872 20J    3172FD 3175 1512 9361SDSR5
D2DM0002010B45920   222 -2100168011 18078371096418059427JN127 80 5927-7865 1736115J P 3174FD-7462-5181 3366SDSR5
D2DM0002010B45920   222 -2100168012 18178370996418659427SP117 68 4209-8261 3746 44J    3   GV 4419 3721 9237SDSR5
D2DM0002010B45920   222 -2100168013 22378368196415459429SP 72 60-2676-8236 5000 14K MB4173FD-8903 4530 2962SDSR5
D2DM0002010B45920   222 -2100168014 22378368196415459429SP 72 60-2676-8236 5000 14K MB4   GV  710 5207 8627SDSR5
D2DM0002010B45920   222 -2100168015 22878367896415159429JN242 20 1606 3020 9397 [illegible]8J BM4         0    0    0SDSR5
D2DM0002010B45920   222 -2100168016 30078362296409759431JN222 20 2542 2269 9397 27K    3163FD 6691-7431    0SDSR5
D2DM0002010B45920   222 -2100168017 32878361196407659432SP 82 85-1386-9865  872 13J M  4171FD-9903 1392    0SDSR5
D2DM0002010B45920   222 -2100168018 32878361196407659432SP 82 85-1386-9865  872 13J M  4   GV  121  863 9962SDSR5
D2DM0002010B45920   222 -2100168019 33378360796407359433SP 92 77  340-9738 2250 16J    3172FD-9856   47 1692SDSR5
D2DM0002010B45920   222 -2100168020 33378360796407359433SP 92 77  340-9738 2250 16J    3   GV  793 2270 9707SDSR5
D2DM0002010B45920   222 -2100168021 34078360396406759433SP 97 80 1200-9775 1736 18J    3166FD -212 1724 9848SDSR5
D2DM0002010B45920   222 -2100168022 34078360396406759433SP 97 80 1200-9775 1736 18J    3   GV -212 1724 9848SDSR5
D2DM0002010B45920   222 -2100168023 38778357196403359435JN102 85 2071-9744  872 20K    3         0    0    0SDSR5
D2DM0002010B45920   222 -2100168024 41078355696401559435SP144 73 7737-5621 2924 48K P  3   GV-1309 3097 9418SDSR5
D2DM0002010B45920   222 -2100168025 41678355296401159436JN 91 69  163-9334 3534 17K    3170FD 9955  486  814SDSR5
D2DM0002010B45920   222 -2100168026 41778355196401059436JN104 92 2396-9659 1392 22J P  3166FD-9695-2292  863SDSR5
D2DM0002010B45920   222 -2100168027 43478354096399059436JN102 74 1599-9403 2756 21J P  3162FD  453 2899 9560SDSR5
D2DM0002010B45920   222 -2100168028 48878350496395359436JN214 14 2006 1353 9703 37L M  4169FD 1668-9807 1022SDSR5
D2DM0002010B45920   222 -2100168029 52078348296393859439CN227 15 1765 1893 9659 34K    4166FD 7314-6820    0SDSR5
D2DM0002010B45920   222 -2100168030 52278348196393259439CN227 15 1765 1893 9659 34K    4166FD 7314-6820    0SDSS3
D2DM0002010B45920   222 -2100168031 61378342096386659442JN156 86 9113-4057  698 25J    3         0    0    0SDSR5
D2DM0002010B45920   222 -2100168032 59778343196387759442JN 79 54-1544-7982 5873 16J MB4178FD 9871-1511  564SDSR5
D2DM0002010B45920   222 -2100168033 58578343996388659441JN107 76 2837-9279 2419 25K    3166FD 9335 3250 1518SDSR5
D2DM0002010B45920   222 -2100168034 61478341996384659442JN107 60 2532-8282 5000 29K    3164FD 9506 3089  302SDSR5
D2DM0002010B45920   222 -2100168035 64778339796383959444JN255 19  802 2965 9511 34L B  4         0    0    0
D2DM0002010B45920   222 -2100168036 66078338996383159444JN 94 80  687-9824 1736 17K    4167FD-9376 -698    0SDSR5
D2DM0002010B45920   222 -2100168037 66978338396382359444JN164 85 9576-2746  872 19K    3160FD-2860-9425 1730SDSR5
D2DM0002010B45920   222 -2100168038 71478335736379159446JN101 87 1505-9303  523 20K    3171FD-9788-1956  870SDSR5
D2DM0002010B45920   222 -2100168039 74778333096376559447JN342 75 7612-5987 2583 57K    3177FD-6412-6298 4385SDSR5
D2DM0002010B45920   222 -2100168040 76078332296375659448CN102 74 1599-9403 2756 21K    4         0    0    0SDSR5
D2DM0002010B45920   222 -2100168041 76878331696375259449CN102 74 1999-9403 2756 21K    4         0    0    0GVLS1
D2DM0002010B45920   222 -2100168042 79978329696372759449SP 95 67  802-9170 3907 19K    3171FD-9983 -460  962SDSR5
D2DM0002010B45920   222 -2100168043 79978329696372759449SP 95 67  802-9170 3907 19K    3   GV  713 3962 9155SDSR5
D2DM0002010B45920   222 -2100168044 82478327996370759450SP 98 81 1375-9781 1964 18K    3175FD-9894-1251  861SDSR5
D2DM0002010B45920   222 -2100168045 82478327996370859450SP 98 81 1375-9781 1564 18K    3   GV 3988 1992 8951SDSR5
D2DM0002010B45920   222 -2100168046 84678326496369259451SP279 77-1524 9624 2250 19J BM2171FD 9877 1564    0SDSR5
D2DM0002010B45920   222 -2100168047 84678326496369259451SP279 77-1524 9624 2250 19J BM2   GV-2216-2551 9412SDSR5
D2DM0002010B45920   222 -2100168048 85678325496368459451JN 88 68 -324-9266 3746 16J    3173FD-9994  349    0SDSR5
D2DM0002010B45920   222 -2100168049 86778325096367659451JN150 80 8529-4924 1736 32K    3159FD-3267-2439 9131SDSR5
D2DM0002010B45920   222 -2100168050 87778324496366959452CN 89 55 -143-8190 5736 19J    3162FD-9998  175    0SDSR5
D2DM0002010B45920   222 -2100168051 87878324396366859452CN 89 55 -143-8190 5736 19J    3         0    0    0GVLS1
D2DM0002010B45920   222 -2100168052 90278322796365059452JN 92 80  344-9942 1736 16J    3         0    0    0SDSR5
D2DM0002010B45920   222 -2100168053 92978320996363059453JN322 70-7405 5785 3420 66J    2164FD 6648 7067 2432SDSR5
D2DM0002010B45920   222 -2100168054 93078320896362959453JN157 85 9170-3992  572 24J    3175FD-3982-8803 2574SDSR5
D2DM0002010B45920   222 -2100168055 95278319396361359454JN102 80 2048-9633 1736 21J    3         0    0    0SDSR5
D2DM0002010B45920   222 -2100168056 96278318796366059455JN 82 75-1384-9565 2538 14J    3         0    0    0SDSR5
D2DM0002010B45920   222 -2100168057 96878315596360159455JN 77 75-2173-9412 2538 13J    3         0    0    0SDSR5
```

```
D2DM0002010R45920   222 -2100168058  98278317[illegible]JN 92 85  34--4654  972 16K   3                0    0    0SDS85
D2DM0002010R45920   222 -2100168059 9817831[illegible]15JN242 25 1219 2205 9659 36K   2                0    C    0SDS85
D2DM0002010R45920   222 -2100168060 98278317316355159455JN207 85 5476 4523  972 10J   3170FD 4000-8508 3407SDS85
D2DM0002010R45920   222 -2100168061 98478317316359059455JN 92 90  349-3994    0 16J   3163FD-9994-349       0SDS85
D2DM0002010R45920   222 -2100168062 99578316196358159456JN 87 90 -523-9986    0 14J   [illegible]      0    0    0SDS85
D2DM0002010R45920   222 -2100168063100078316196357759456JN197 30 9419 2879 1736 11I   2                0    0    0SDS85
D2DM0002010R45920   222 -2100168064100178316196357759456JN247 .5  341 802 9962 88K   2167FD 9205-3907       0SDS85
D2DM0002010R45920   222 -2100168065    07838309643205942 1DG237 20 1863 2868 9397 28 D                  0    0    0SDS85
D2DM0002010R45920   222 -2100168066    07838309643205942 1DG187 20 3395  417 9397 32 D                  0    0    0SDS85
D2DM0002010R45920   222 -2100168067    07838339643205942 1DG247 10  679 1598 9848 52 D                  0    0    0SDS85
D2DM0002010R45920   222 -2100168068    07838309643205942 1DG222 20 2542 2289 9397 27 D                  0    C    0SDS85
```

SECTION 3: SYSTEM DOCUMENTATION

COMPUTER EQUIPMENT:

CPU:IBM 360 model 67; 32-bit word.

Memory: IBM 2365-2; 1 million bytes.

PERIPHERAL EQUIPMENT:

IBM 1403N1 printers; IBM 2501-B2 card readers; IBM 2540 card reader/punch; IBM 2301 drums; IBM 3333 and 3330 disc drives; IBM 3420-5 9-track dual-density tape drives (800/1600 bpi); IBM 2741 communications terminals; IBM 3277 display terminals; Calcomp 770/665 drum plotter (paper width 29 5/16 inches; pen speed 2.25 inches per second; increment sizes 0.0025 and 0.005 inches).

OPERATING SYSTEM:

Michigan Terminal System; a time-sharing virtual-memory system.

PROGRAM LISTING:

```
      REAL*8 AID
      REAL*4 LINTYP,      VT(3),VR(3),RM(9),BL/'    '/,RM2(9)
      INTEGER*4 DSCALE,GSCALE,ESCALE
      EQUIVALENCE (COSA,VT),(COSB,VT(2)),(COSG,VT(3))
      LOGICAL*1 NOCOOR
C
C              READ GRID SCALE FACTOR AND ELEVATION SCALE FACTOR
C
      READ(5,802)GSCALE,ESCALE
      NCD=0
C
C        READ TYPE 1 CARD
C
  100 READ(1,800,END=900)AID,IELEV,ADOM,AFM,ABLVL,ABLOC,IEAST,INORTH,
     *ITT,ITP,    ALEN,DSCALE,NOBS,IREF
      WRITE(6,805)        AID,IELEV,ADOM,AFM,ABLVL,ABLOC,IEAST,INORTH,
     *ITT,ITP,ALEN,DSCALE,NOBS,IREF
      NCD=NCD+1
      NOCOOR=.FALSE.
      IF((INORTH.EQ.0).OR.(IEAST.EQ.0).OR.(IELEV.EQ.0))NOCOOR=.TRUE.
C
C        CONVERT ALL COORDINATES TO BASIC UNIT
C
      RELEV=IELEV
      IF(ESCALE.NE.0)RELEV=IELEV*10.0**ESCALE
      REAST=IEAST
      RNORTH=INORTH
      IF(GSCALE.EQ.0)GOTO 105
      REAST=IEAST*10.0**GSCALE
      RNORTH=INORTH*10.0**GSCALE
  105 CONTINUE
C
C              COMPUTE DIRECTION COSINES OF TRAVERSE
C
      T=ITT*1.745329E-2
      P=ITP*1.745329E-2
      ST=SIN(T)                                                         CNVR 370
      CT=COS(T)                                                         CNVR 380
      SP=SIN(P)                                                         CNVR 390
      CP=COS(P)                                                         CNVR 400
      X=CT*CP                                                           CNVR 410
      Y=ST*CP                                                           CNVR 420
      Z=SP                                                              CNVR 430
C                                                                       CNVR 440
C           IF BOREHOLE DATA, COMPUTE ROTATION MATRIX                   CNVR 450
C                                                                       CNVR 460
      IF(IREF.NE.999)GOTO 120                                           CNVR 470
      RM(1)=CT*SP                                                       CNVR 480
      RM(2)=ST*SP                                                       CNVR 490
      RM(3)=-CP                                                         CNVR 500
      RM(4)=-ST                                                         CNVR 510
      RM(5)=CT                                                          CNVR 520
      RM(6)=0.0                                                         CNVR 530
      RM(7)=X                                                           CNVR 540
      RM(8)=Y                                                           CNVR 550
      RM(9)=Z                                                           CNVR 560
C                                                                       CNVR 570
C           READ TYPE 2 CARDS                                           CNVR 580
C                                                                       CNVR 590
  120 DO 1000 LOOP=1,NOBS                                               CNVR 600
C                                                                       CNVR 610
      READ(1,804)IDIST,ATYPE,IAZ,IDIP,      ASIZE,ASPAC,AFILL1,         CNVR 620
     *AFILL2,AWAT,AWAV,LINTYP,IPITCH,ALITH,AHDN                         CNVR 630
      NCD=NCD+1                                                         CNVR 640
      IF(NOCOOR)GOTO 128                                                CNVR 650
      IF(IDIST.LE.0)GOTO 128                                            CNVR 660
C                                                                       CNVR 670
C           CONVERT DISTANCE TO BASIC UNIT                              CNVR 680
C                                                                       CNVR 690
      DIST=IDIST                                                        CNVR 700
      IF(DSCALE.NE.0)DIST=IDIST*10.0**DSCALE                            CNVR 710
C                                                                       CNVR 720
```

```
C                       INCREMENT COORDINATES
C
      SNORTH=RNORTH+X*DIST
      SEAST=REAST+Y*DIST
      SELEV=RELEV-Z*DIST
      JNORTH=SNORTH+SIGN(0.5,SNORTH)
      JEAST=SEAST+SIGN(0.5,SEAST)
      JELEV=SELEV+SIGN(0.5,SELEV)
      GOTO 130
  128 JNORTH=RNORTH+SIGN(0.5,RNORTH)
      JEAST=REAST+SIGN(0.5,REAST)
      JELEV=RELEV+SIGN(0.5,RELEV)
  130 IF(IREF.EQ.999) GOTO 250
C
C                     COMPUTE DIP DIRECTION AND DIP OF DISCONTINUITY
C
      IAZ=IAZ+IREF
      IF(IAZ.GT.360) IAZ=IAZ-360
      IF(IDIP.GE.0) GO TO 140
      IAZ=IAZ-180
      IF(IAZ.LT.0) IAZ=IAZ+360
      IDIP=-IDIP
  140 CONTINUE
C
C                     COMPUTE DIRECTION COSINES OF DISCONTINUITY
C
      T=(IAZ+180)*1.745329E-2
      P=(90-IDIP)*1.745329E-2
      CP=COS(P)
      COSA=COS(T)*CP
      COSB=SIN(T)*CP
      COSG=SIN(P)
      GOTO 300
C
C                       COMPUTE DIRECTION COSINES OF DISCONTINUITY IN RECOR
C                       ORIENTATION
```

```
C
  250 T=(IAZ+180)*1.745329E-2
      P=(90-IDIP)*1.745329E-2
      CP=COS(P)
      VR(1)=COS(T)*CP
      VR(2)=SIN(T)*CP
      VR(3)=SIN(P)
C
C                   COMPUTE TRUE DIRECTION COSINES OF DISCONTINUITY BY
C                   ROTATION
C
      CALL GMPRD(RM,VR,VT,3,3,1)
C
C                   COMPUTE TRUE DIP DIRECTION AND DIP OF DISCONTINUITY
C
      CALL DCAN(COSA,COSB,COSG,1,T,P,1,-2,1)
      IAZ=T+0.5
      IDIP=P+0.5
C
C                  COMPUTE TERZAGHI WEIGHTING FACTOR
C
  300 CONTINUE
      CALPHA=ABS(COSA*X+COSB*Y+COSG*Z)
      IF(CALPHA.LT.0.0872)CALPHA=0.0872
      W=1.0/CALPHA
C
C                  COMPUTE INTEGRAL VALUES FOR OUTPUT
C
      ICOSA=IDC(COSA)
      ICOSB=IDC(COSB)
      ICOSG=IDC(COSG)
      IW=W*10.0+0.5
C
C            COMPUTE DIRECTION COSINES OF LINE
C
      IF(LINTYP.EQ.BL)GOTO 340
      IF(IPITCH.EQ.0)GOTO 330                                           CNVR1450
      TAX=IAZ*1.745329E-2                                               CNVR1460
      PAX=(IDIP-90)*1.745329E-2                                         CNVR1470
      ANG=IPITCH*1.745329E-2                                            CNVR1480
      CALL  ROTMAT(TAX,PAX,ANG,RM2)                                     CNVR1490
      VT(1)=SIN(TAX)                                                    CNVR1500
      VT(2)=-COS(TAX)                                                   CNVR1510
      VT(3)=0.0                                                         CNVR1520
      CALL  GMPRD(RM2,VT,VR,3,3,1)                                      CNVR1530
      LINA=IDC(VR(1))                                                   CNVR1540
      LINB=IDC(VR(2))                                                   CNVR1550
      LING=IDC(VR(3))                                                   CNVR1560
      GOTO  400                                                         CNVR1570
  330 LINTYP=BL                                                         CNVR1580
  340 LINA=0                                                            CNVR1590
      LINB=0                                                            CNVR1600
      LING=0                                                            CNVR1610
  400 CONTINUE                                                          CNVR1620
C                                                                       CNVR1630
C                   WRITE OUTPUT RECORD                                 CNVR1640
C                                                                       CNVR1650
      WRITE(7,801)AID,ADOM,AFM,ABLVL,ABLOC,ITT,ITP,ALEN,NOBS,DSCALE,    CNVR1660
     *LOOP,                                                             CNVR1670
     *IDIST,JEAST,JNORTH,JELEV,ATYPE,IAZ,IDIP,ICOSA,ICOSB,ICOSG,IW,ASIZECNVR1680
     *,ASPAC,AFILL1,AFILL2,AWAT,AWAV,LINTYP,LINA,LINB,LING,ALITH,AHDN   CNVR1690
 1000 CONTINUE                                                          CNVR1700
      GOTO 100                                                          CNVR1710
C                                                                       CNVR1720
C                                                                       CNVR1730
C                                                                       CNVR1740
  800 FORMAT(6X,A7,1X,I5,1X,A2,1X,A3,1X,A4,1X,A3,2(1X,I6),2(1X,I3),     CNVR1750
     *         1X,A4,1X,I1,1X,I2,1X,I3)                                 CNVR1760
  801 FORMAT('D',A7,A2,A3,A4,A3,2I3,A4,I2,I1,I2,I4,2I6,I5,A2,2I3,       CNVR1770
     *3I5,I3,5A1,A3,A2,3I5,A3,A2)                                       CNVR1780
  802 FORMAT(2I1)                                                       CNVR1790
  804 FORMAT(6X,I4,1X,A2,2(1X,I3),        5(1X,A1),1X,A3,1X,A2,1X,I3,   CNVR1800
     *1X,A3,1X,A2)                                                      CNVR1810
  805 FORMAT(6X,A7,1X,I5,1X,A2,1X,A3,1X,A4,1X,A3,2(1X,I6),2(1X,I3),     CNVR1820
     *1X,A4,1X,I1,1X,I2,1X,I3)                                          CNVR1830
C                                                                       CNVR1840
C                                                                       CNVR1850
  900 ENDFILE 7                                                         CNVR1860
      STOP                                                              CNVR1870
C                                                                       CNVR1880
C                                                                       CNVR1890
      END                                                               CNVR1900
C                                                                       CNVR1910
C                                                                       CNVR1920
      FUNCTION IDC(R)                                                   CNVR1930
      IDC=R*10000.0+SIGN(0.5,R)                                         CNVR1940
      IF(IABS(IDC).GT .9999) IDC=ISIGN(9999,IDC)                        CNVR1950
      RETURN                                                            CNVR1960
      END                                                               CNVR1970
      SUBROUTINE DCAN(A,B,G,J1,T,P,J2,J3,N)                             CNVR1980
      REAL A(N),B(N),G(N),T(N),P(N),RTD/57.29578/                       CNVR1990
      IF(N.LT.1) RETURN                                                 CNVR2000
      I1=IABS(J1)                                                       CNVR2010
      I2=IABS(J2)                                                       CNVR2020
      K1=1-I1                                                           CNVR2030
      K2=1-I2                                                           CNVR2040
      DO 100 I=1,N                                                      CNVR2050
      K1=K1+I1                                                          CNVR2060
      K2=K2+I2                                                          CNVR2070
   10 AA=A(K1)                                                          CNVR2080
      BB=B(K1)                                                          CNVR2090
      GG=G(K1)                                                          CNVR2100
C                  ADJUST OUTPUT TO REQUIRED DOMAIN                     CNVR2110
      IF(J3) 13,13,15                                                   CNVR2120
   13 IF(GG) 14,15,15                                                   CNVR2130
   14 AA=-AA                                                            CNVR2140
      BB=-BB                                                            CNVR2150
      GG=-GG                                                            CNVR2160
   15 PL=ARSIN(GG)                                                   CNVR2170
      IF(AA)18,20,18                                                 CNVR2180
   18 TR=ATAN(BB/AA)                                                 CNVR2190
      IF(TR.LT.0.0)TR=TR+3.14159265                                  CNVR2200
      IF(BB.EQ.0.0 .AND. AA.LT.0.0)TR=3.14159265                     CNVR2210
      GOTO 30                                                        CNVR2220
   20 TR=1.57079633                                                  CNVR2230
   30 IF(BB.LT.0.0)TR=TR+3.14159265                                  CNVR2240
C                     ADJUST OUTPUT TO REQUIRED TYPE                 CNVR2250
      I3=IABS(J3)                                                    CNVR2260
      GOTO (38,36,34),I3                                             CNVR2270
   34 TR=TR-1.570796                                                 CNVR2280
   36 TR=TR-3.141593                                                 CNVR2290
      IF(TR.LT.0.0)TR=TR+6.283185                                    CNVR2300
      PL=1.570796-PL                                                 CNVR2310
C                     ADJUST OUTPUT TO REQUIRED UNITS                CNVR2320
   38 IF(J2)40,40,50                                                 CNVR2330
   40 T(K2)=TR                                                       CNVR2340
      P(K2)=PL                                                       CNVR2350
      GOTO 100                                                       CNVR2360
   50 T(K2)=TR*RTD                                                   CNVR2370
      P(K2)=PL*RTD                                                   CNVR2380
  100 CONTINUE                                                       CNVR2390
      RETURN                                                         CNVR2400
      END                                                            CNVR2410
      SUBROUTINE ROTMAT(A,B,C,T)                                     CNVR2420
C THIS SUBROUTINE COMPUTES A 3X3 ROTATION MATRIX GIVEN AN AXIS OF    CNVR2430
C ROTATION AND AN ANGLE OF ROTATION. A,B ARE THE TREND AND PLUNGE OF CNVR2440
C THE AXIS OF ROTATION AND C IS THE ANGLE OF ROTATION (ALL IN RADIANS) CNVR2450
C AND T IS THE RETURNED MATRIX.                                      CNVR2460
      REAL T(9),P(9),Q(9),R(9)                                       CNVR2470
      SINA=SIN(A)                                                    CNVR2480
      COSA=COS(A)                                                    CNVR2490
      SINB=SIN(B)                                                    CNVR2500
      COSB=COS(B)                                                    CNVR2510
      SINC=SIN(C)                                                    CNVR2520
```

```
COSC=COS(C)                                                 CNVR2530
Q(1)=COSA*COSB                                              CNVR2540
Q(2)=-SINA                                                  CNVR2550
Q(3)=-COSA*SINB                                             CNVR2560
Q(4)=SINA*COSB                                              CNVR2570
Q(5)=COSA                                                   CNVR2580
Q(6)=-SINA*SINB                                             CNVR2590
Q(7)=SINB                                                   CNVR2600
Q(8)=0.                                                     CNVR2610
Q(9)=COSB                                                   CNVR2620
P(1)=1.                                                     CNVR2630
P(2)=0.                                                     CNVR2640
P(3)=0.                                                     CNVR2650
P(4)=0.                                                     CNVR2660
P(5)=COSC                                                   CNVR2670
P(6)=SINC                                                   CNVR2680
P(7)=0.                                                     CNVR2690
P(8)=-SINC                                                  CNVR2700
P(9)=COSC                                                   CNVR2710
CALL GMPRD(P,Q,R,3,3,3)                                     CNVR2720
CALL GTPRD(R,Q,T,3,3,3)                                     CNVR2730
RETURN                                                      CNVR2740
END                                                         CNVR2750
```

VARIABLES:

Input variables:

Parameter card:

GSCALE, ESCALE	scale factors for grid coordinates and elevation respectively.

Type 1 card:

AID	traverse number (A)
IELEV, IEAST, INORTH	elevation, easting and northing of traverse origin
ADOM	domain (A)
AFM	formation (A)
ABLVL	bench level (A)
ABLOC	bench location (A)
ITT, ITP	trend and plunge of traverse
ALEN	length of traverse (A)
DSCALE	scale factor for distance of traverse
NOBS	number of observations on traverse
IREF	reference direction

Type 2 card:

IDIST	distance along traverse
ATYPE	discontinuity type (A)
IAZ, IDIP	recorded azimuth and dip
ASIZE	size code for discontinuity (A)
ASPAC	spacing of discontinuities (A)
AFILL1, AFILL2	filling codes (A)

AWAT, AWAV	codes for water and waviness (A)
LINTYP	type of line, if any (A)
IPITCH	pitch of line
ALITH, AHDN	codes for lithology and hardness (A)

Output:

LOOP	counter for observations on each traverse
JEAST, JNORTH, JELEV	easting, northing and elevation of point of observation rounded to integers
IAZ, IDIP	dip direction and dip of discontinuity
ICOSA, ICOSB, ICOSG	direction cosines of normal to discontinuity, times 10^4 and rounded to integers.
IW	terzaghi correction factor, times 10 and rounded.
LINA, LINB, LING	direction cosines of linear, times 10^4 and rounded to integers.

Internal:

NCD	counter for number of cards read.
NOCOOR	flag for missing coordinates - input
RELEV, REAST, RNORTH	elevation, easting and northing of traverse origin.
T, P	trend and plunge of traverse - radians; trend and plunge of normal to discontinuity in radians; dip direction and dip of discontinuity in degrees
ST, CT	sine and cosine of trend of traverse

SP, CP	sine and cosine of plunge of traverse
X, Y, Z	direction cosines of traverse
RM	rotation matrix to rotate reference axes north, east and down into the axes U, V and W, where W is the borehole direction and V is horizontal
DIST	distance from traverse origin
SNORTH, SEAST, SELEV	northing, easting and elevation of point of observation
COSA, COSB, COSG	direction cosines of normal to discontinuity
VR	vector containing direction cosines of normal to discontinuity in recorded orientation; vector containing the direction cosines of the strike vector.
VT	vector containing direction cosines of normal to discontinuity - true orientation; vector containing direction cosines of linear.
CALPHA	cosine of angle between normal to discontinuity and traverse or borehole.
W	terzaghi correction factor for discontinuity
TAX, PAX, ANG	trend and plunge of axis of rotation and angle of rotation, in radians, to rotate strike vector through pitch angle into coincidence with linear.
RM2	rotation matrix to rotate the strike vector through the pitch angle into coincidence with the linear.

Equivalences:

COSA, COSB and COSG occupy the same storage locations as the vector VT.

Subroutines and Function Subprograms:

Function IDC(R): takes the direction cosine R, multiplies it by 10^4 and rounds it to the nearest integer. A maximum absolute value of 9999 is imposed. The result is returned as the function value.

Subroutine GMPRD: from the SSP subroutine package (IBM). Multiplies two matrices.

Calling sequence:

```
CALL GMPRD (A,B,C,I,J,K)
```

The J*K matrix B is premultiplied by the I*J matrix A and the result placed - the I*K matrix C.

Subroutine DCAN: included in subroutine library accompanying these programs. Calling sequence -

```
CALL DCAN (A,B,G,I,T,P,J,K,N)
```

The vectors T and P of angles are computed from the vectors, A,B, and G of direction cosines. I is the spacing of direction cosines - A, B and G, and J the spacing of angles in T and P. For example, if J=2, output angles are placed in T(1), T(3), T(5), etc. If |K|=1, trends and plunges are computed; if |K|=2, dip directions and dips assuming the input directions are normals to planes; if |K|=3, strikes and dips are computed. If K>0, angles are returned as radians; if K<0, angles are returned as degrees. N is the number of directions given.

Subroutine ROTMAT: Accompanying subroutine library. Computes a rot-matrix for rotation through a given angle about a given axis. Calling sequence:

```
CALL ROTMAT (A,B,C,R)
```

A,B, and C are the trend and plunge of the axis of rotation and the angle of rotation, in radians R is the rotation matrix.

DATA STRUCTURES: The input data file consists of unblocked 80-byte records. The program does not block output records. Output data records are 112 bytes long. One output data record is written for each type 2 input record.

STORAGE REQUIREMENTS:

Main program	4232
Function IDC	416
Subroutine GMPRD	680
Subroutine DCAN	1344
Subroutine ROTMAT	848
Total	7520 bytes.

MAINTENANCE AND UPDATES: None

SECTION 4: OPERATING DOCUMENTATION

OPERATOR INSTRUCTIONS: None

OPERATING MESSAGES: None

CONTROL CARDS:

```
$RUN *MOUNT PAR=1234 9TP *TAPE* VOL=ABCDEF FMT=FB(5600,112) RING=IN
$RUN *SOURCE* 1=*SOURCE* 5=*SOURCE* 6=*PRINT* 7=*TAPE*
[parameter card
┌
•
• data cards
•
└
$ENDFILE
```

ERROR RECOVERY: No special procedures.

RUN TIME: Approximately 20 seconds CPU time per 1000 observations.

Identification

Program name:

DATFIX1

Program title:

Areal-analysis package - data-file conditioning program.

Author:

John Ramsden

Organization:

Department of Geology, University of Alberta, Edmonton, Alberta, Canada

Date:

5 October 1974

Program version number:

1

Application

Purpose:

To make adjustments to a data file so that it is in a suitable form for areal analysis.

Method:

The fabric data are examined one traverse at a time. Those observations that have no distance associated with them are "distributed" uniformly along the traverse. The algorithm used for doing this is as follows.
l=length of traverse
n=number of observations without distance recorded
x(i)=distance assigned to the i-th observation without distance

a=l/n

```
t=a/2

x(i)=(i*a)-t :i=1,2,...n
```

All the observations on the traverse are then ordered by increasing distance, and all the data for that traverse are written on the output file. The next traverse is then read from the input file and the process repeated.

The first observation with zero distance following at least one observation with non-zero distance is taken as the first observation for which distance is not recorded. This observation and all those that follow it in the traverse are assumed to have no distance recorded.

Data restrictions:

Maximum number of observations in one traverse is 100. Those observations without distance must follow those with distance in each traverse.

Timing:

The program requires about 2 seconds of CPU time for each 100 lines in the data file.

Implementation

Original installation:

Organization: University of Alberta Computing Services.
Computer: IBM 360 model 67 (1024K).
I/O devices: card, disc, tape, typewriter terminals and display terminals.
Software: Michigan Terminal System (MTS), a virtual-storage time-sharing system.

Source language:

FORTRAN IV

Subroutines required: None.

Size:

13518 bytes.

Requirements for I/O devices:

I/O unit no.	read/ write	Maximum FORTRAN record length (bytes)	device control statements	suggested device
1	read	112 , formatted	none	tape; disc
2	write	112 , formatted	none	tape; disc
6	write	37 , formatted	none	line printer

Usage

Input:

I/O unit 1: Input data file

The records of the data file must all have the same length and format. 28 variables are read in each record with the format
(A4,I4,3A4,2I3,I4,A4,A1,I4,2I6,I5,14A4).
Those fields which are used by the program are as follows.

Variable no.	Columns	Specifications
2	005-008	integer, traverse number
6	021-023	integer, trend of traverse
7	024-026	integer, plunge of traverse
8	027-030	integer, length of traverse
11	036-039	integer, distance along traverse
12	040-045	integer, easting
13	046-051	integer, northing
14	052-056	integer, elevation

Output:

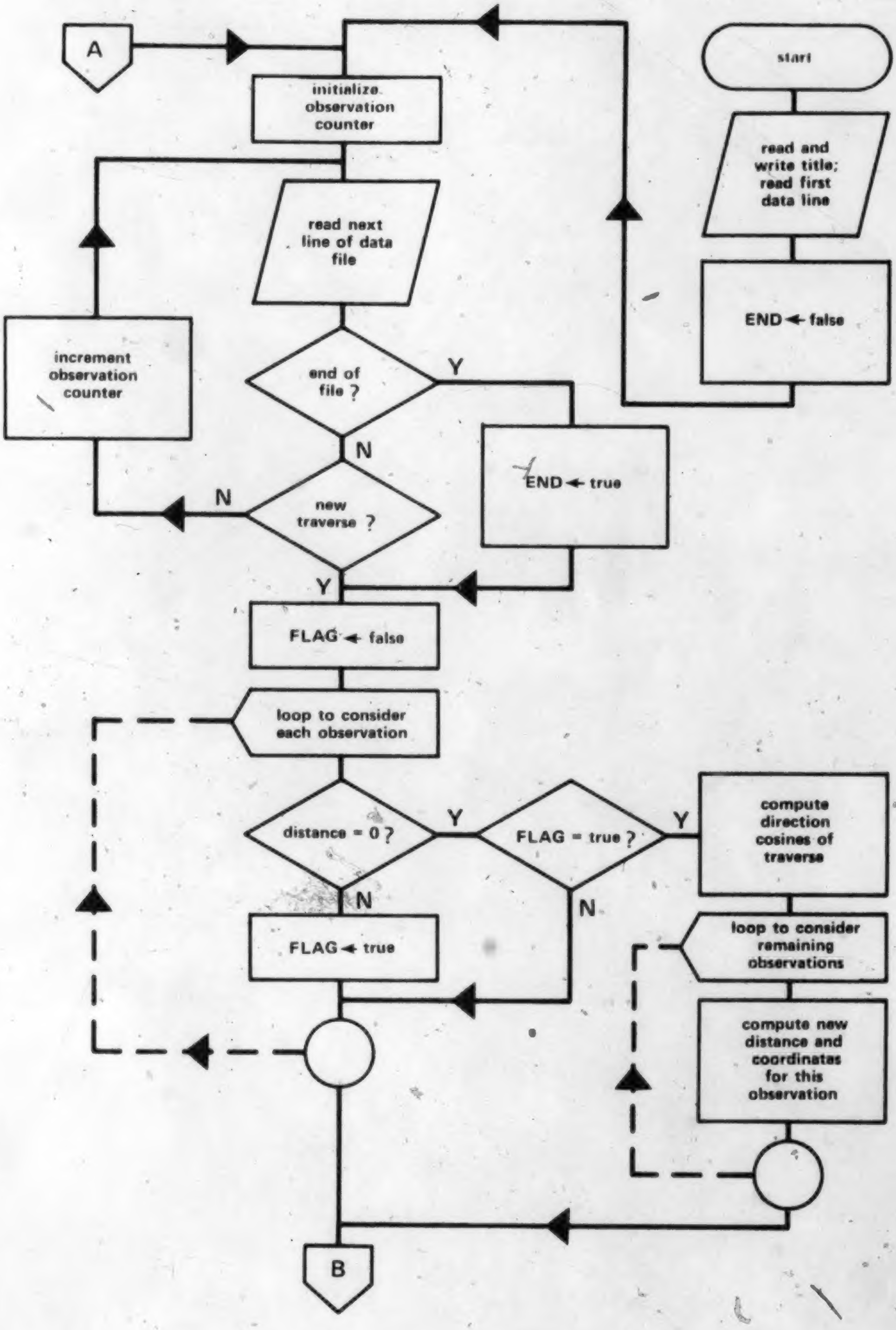
I/O unit 2: Output data file

Description same as input data file.

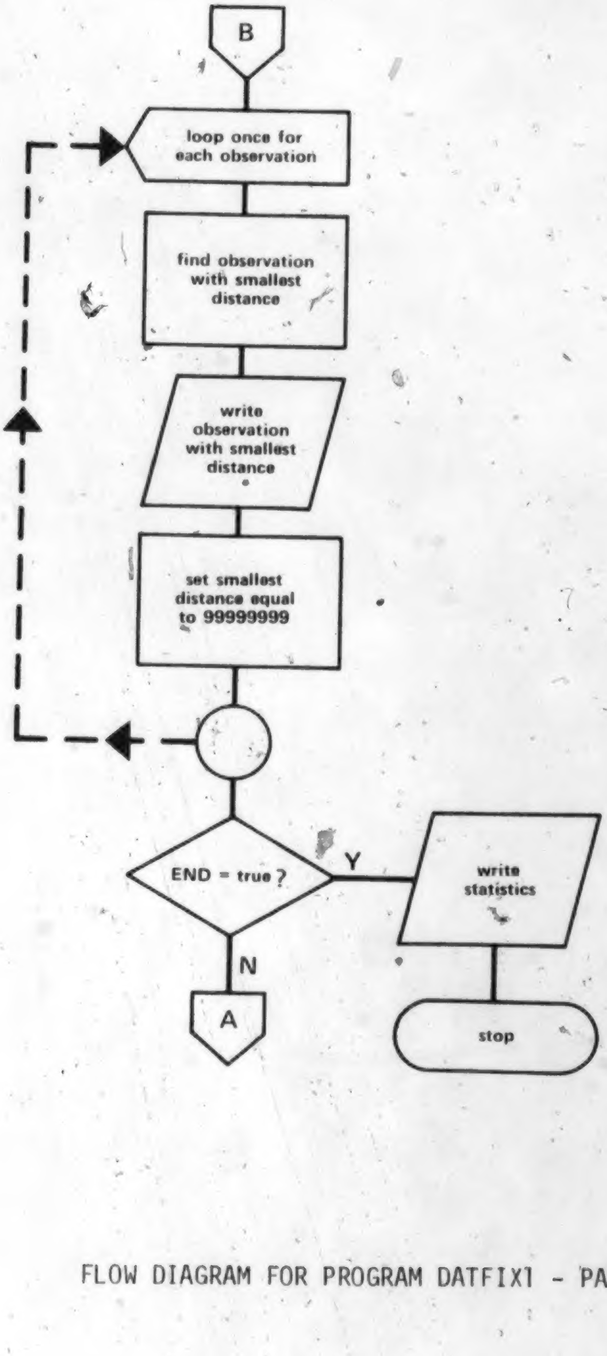
I/O unit 6: Statistics

On this unit the program writes the number of traverses processed and the number of observations that were without distance.

```
C
C
C AREAL-ANALYSIS PACKAGE - PROGRAM "DATFIX1".
C
C
      LOGICAL*1 END,FLAG
      INTEGER   K(28,100),TITLE(30)
      REAL DTR/1.745329E-2/
C
    1 FORMAT(A4,I4,3A4,2I3,I4,A4,A1,I4,2I6,I5,14A4)
    2 FORMAT('T',30A4)
    3 FORMAT(/' PROGRAM "PFIX" STATISTICS'/' NUMBER OF TRAVERSES: ',I4/
     *' OBSERVATIONS WITHOUT DISTANCE: ',I5)
C
C-----READ & WRITE TITLE
C
      READ(1,2)TITLE
      WRITE(2,2)TITLE
C
C-----READ FIRST DATA LINE
C
      NTOT=0
      NTRAV=0
      READ(1,1)(K(I,1),I=1,28)
      END=.FALSE.
   90 J=2
C
C-----READ NEXT LINE
C
  100 READ(1,1,END=200)(K(I,J),I=1,28)
      IF(K(2,J).EQ.0)GOTO200
      IF(K(2,J).NE.K(2,J-1))GOTO210
      J=J+1
      GOTO100
C
C-----END OF TRAVERSE
C
  200 END=.TRUE.
  210 NO=J-1
C
C-----FIND FIRST OBSERVATION WITH DISTANCE MISSING
C
      J=1
      FLAG=.FALSE.
  260 IF(K(11,J).EQ.0)GOTO300
      FLAG=.TRUE.
  280 J=J+1
      IF(J-NO)260,260,400
C
  300 IF(FLAG)GOTO360
      GOTO280
C
C-----COMPUTE DIRECTION COSINES OF TRAVERSE
C
  360 CONTINUE
      TR=K(6,1)*DTR
      PL=K(7,1)*DTR
      CP=COS(PL)
      DNORTH=COS(TR)*CP
      DEAST=SIN(TR)*CP
      DELEV=-SIN(PL)
C
C-----COMPUTE NEW DISTANCES AND COORDINATES
C
      NFIRST=J
      NZ=NO-NFIRST+1
      A=K(8,1)/FLOAT(NZ)
      B=A/2.0
      I=0
      DO 380 J=NFIRST,NO
      NTOT=NTOT+1
      I=I+1
      X=I*A-B
      K(11,J)=X+0.5
      K(12,J)=K(12,J)+X*DEAST+0.5
      K(13,J)=K(13,J)+X*DNORTH+0.5
      K(14,J)=K(14,J)+X*DELEV+0.5
  380 CONTINUE
C
C-----WRITE LINES IN ORDER OF INCREASING DISTANCE
C
  400 CONTINUE
      NTRAV=NTRAV+1
      DO 420 L=1,NO
      MIN=99999999
      DO 410 M=1,NO
      IF(K(11,M).GE.MIN)GOTO410
      MIN=K(11,M)
      J=M
  410 CONTINUE
      WRITE(2,1) (K(I,J),I=1,28)
      K(11,J)=99999999
  420 CONTINUE
      IF(END)GOTO600
      J=NO+1
      DO 500 I=1,28
  500 K(I,1)=K(I,J)
      GOTO90
C
  600 WRITE(6,3)NTRAV,NTOT
      STOP
      END
```


A
initialize observation counter
start
read and write title; read first data line
END ← false
read next line of data file
increment observation counter
end of file ?
Y
N
END ← true
new traverse ?
N
Y
FLAG ← false
loop to consider each observation
distance = 0 ?
Y
FLAG = true ?
Y
compute direction cosines of traverse
N
N
FLAG ← true
loop to consider remaining observations
compute new distance and coordinates for this observation
B

FLOW DIAGRAM FOR PROGRAM DATFIX1 - PART 1


B
loop once for each observation
find observation with smallest distance
write observation with smallest distance
set smallest distance equal to 99999999
END = true ?
Y
N
A
write statistics
stop

FLOW DIAGRAM FOR PROGRAM DATFIX1 - PART 2

Identification

Program name:

CDKEY1

Program title:

Areal-analysis package - merging of keys.

Author:

John Ramsden

Organization:

Department of Geology, University of Alberta, Edmonton, Alberta, Canada

Date:

1 October 1974

Program version number:

1

Application

Purpose:

To read several keys (sets of integers) and merge them into a single key.

Method:

The program reads sets of integers and constructs a new set containing all the integers in all the sets read. Not all the sets in the input file are used. Which sets are used is determined by a set of sequence sumbers read from another I/O unit.

The integers in the input sets are copied sequentially into the output set as they are read. After all the input sets have been copied, the integers in the output set are ordered by calling the subroutine ORDERH.

The program does not check for duplicates. An integer will appear in the output set as many times as it appears in all of the input sets.

Data restrictions:

Maximum number of input sets to be included is 100. Maximum size of each input set is 20 integers. Maximum size of the output set is 1000 integers.

Timing:

The program required 1.3 seconds of CPU time to merge 11 keys out of a file containing 32 keys. Average size of the keys was 10 line numbers.

Implementation

Original installation:

Organization:	University of Alberta Computing Services.
Computer:	IBM 360 model 67 (1024K).
I/O devices:	card, disc, tape, typewriter terminals and display terminals.
Software:	Michigan Terminal System (MTS), a virtual-storage time-sharing system.

Source language:

FORTRAN IV

Subroutines required:

Subroutines READK, WRITEK, ORDERH from JR subroutine library.

Size:

Total program size including all required subroutines is 5704 bytes.

Requirements for I/O devices:

I/O unit no.	read/ write	Maximum FORTRAN record length (bytes)	device control statements	suggested device
1	read	80 formatted	none	tape; disc; card reader.
2	read	80 formatted	none	tape; disc; card reader.
7	write	80 formatted	none	tape; disc; card punch.

Usage

Input:

I/O unit 1: input sets.

The format of each set is as written by the subroutine WRITEK.

I/O unit 2: numbers of those input sets to be used.

Same format as the input sets.

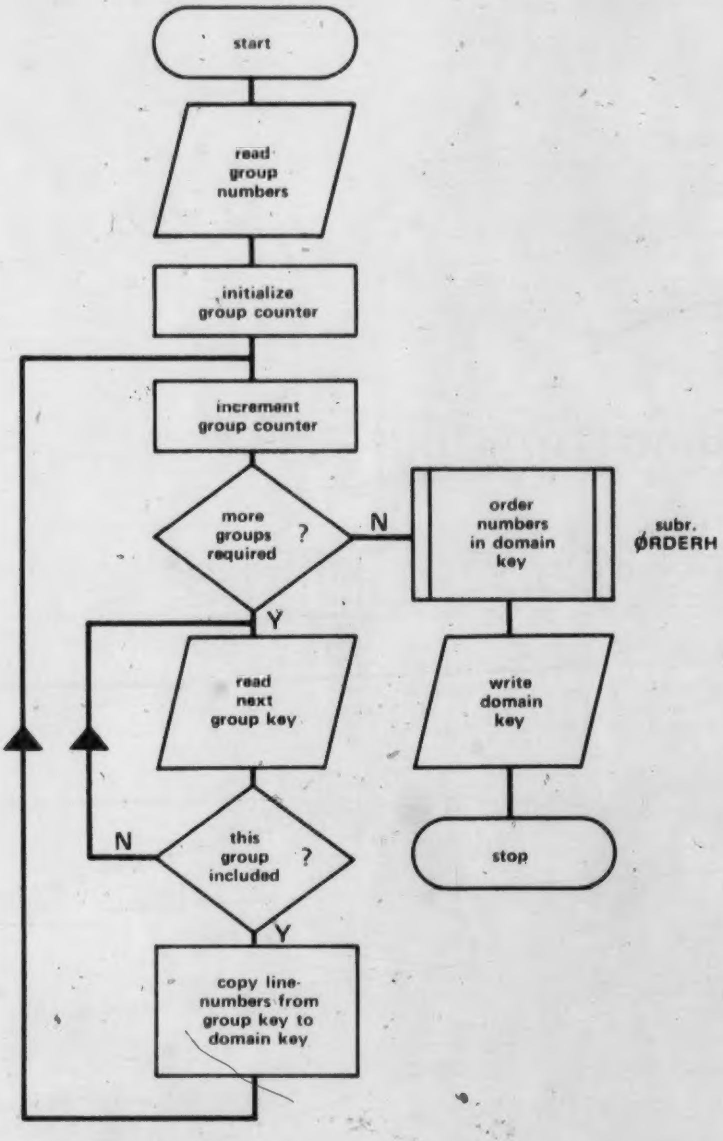
Output:

I/O unit 7: output set.

Written by subroutine WRITEK. Same format as input sets.

L

```
C
C
C AREAL-ANALYSIS PACKAGE - PROGRAM "DDKEY1".
C
C
      INTEGER*2 KEY(20),MKEY(1000),IGN(100)
      INTEGER TITLE(30),OV(5)/5*'    '/,TITLE2(30)
      LEN=0
      CALL READK(2,TITLE2,NG,NGD,IGN,&410,&420)
      DO 40 I=1,5
   40 TITLE2(10+I)=QV(I)
      IG=0
      IGF=0
   60 IGF=IGF+1
      IF(IGF.GT.NGD)GOTO300
   80 IG=IG+1
      CALL READK(1,TITLE,LENFIL,LENK,KEY,&300,&400)
      IF(IG.NE.IGN(IGF))GOTO80
      DO 200 J=1,LENK
      LEN=LEN+1
  200 MKEY(LEN)=KEY(J)
      GOTO60
  300 CALL ORDERH(MKEY,LEN)
      CALL WRITEK(7,TITLE2,LENFIL,LEN,MKEY)
      STOP
  400 STOP 12
  410 STOP11
  420 STOP10
      END
```


start
read group numbers
initialize group counter
increment group counter
more groups required ?
N
Y
order numbers in domain key
subr. ØRDERH
write domain key
stop
read next group key
this group included ?
N
Y
copy line-numbers from group key to domain key

FLOW DIAGRAM FOR PROGRAM DDKEY1

Identification

Program name:

DSEM

Program title:

Density estimator testing program.

Author:

John Ramsden

Organization:

Department of Geology, University of Alberta, Edmonton, Alberta, Canada

Date:

1 October 1974

Program version number:

1

Application

Purpose:

To compute density estimates on the sphere by several methods.

Method:

The program computes density estimates on the sphere by several methods at several counting locations for each sample of data points given. For details of the methods see Ramsden J, 1975, Numerical Methods in Fabric Analysis, Ph D thesis, Univ of Alberta, Edmonton.

Data restrictions:

Maximum number of counting locations is 20. Maximum size of samples is 1000 points.

Timing:

About 30 seconds of CPU time was required to process 50 samples of 50 points each using 6 counting locations and computing 10 density estimates at each location.

Implementation

Original installation:

Organization: University of Alberta Computing Services.
Computer: IBM 360 model 67 (1024K).
I/O devices: card, disc, tape, typewriter terminals and display terminals.
Software: Michigan Terminal System (MTS), a virtual-storage time-sharing system.

Source language:

FORTRAN IV

Subroutines required:

None.

Size:

Total program size is 19208 bytes.

Requirements for I/O devices:

I/O unit no.	read/ write	Maximum FORTRAN record length	device control statements	suggested device
1	read	12128 unformatted	rewind	tape, disc
4	read	43 formatted	none	card reader
5	read	80 formatted	none	card reader
6	write	69 formatted	none	line printer
8	write	124 formatted	none	tape, disc

Usage

Input:

I/O unit 1: samples.

Any number of samples. One unformatted record for each sample, each record consisting of 4 bytes containing the number of points in the sample as a fullword integer, 120 bytes containing the title, 4 bytes containing the number of direction cosines in the record, and 12 to 12000 bytes containing the direction cosines of the data points as one-word real numbers.

I/O unit 4: counting locations.

One record containing the number of counting locations in columns 1-4 as a right-justified integer, followed by one record for each counting location. Each of these records contains

col	contents
1-3	number of this counting location
4-13	first direction cosine
14-23	second direction cosine
24-33	third direction cosine
34-43	true density at this counting location

I/O unit 5: parameters.

One record for each density estimate to be computed. Each record contains

col	contents
1	1 for the constant-area estimator; 2 for the constant-area estimator using Kamb's rule; 3 for the constant-proportion estimator; 4 for the Fisher-weighted estimator.
2	blank
3-11	parameter of the estimator as a real number (with decimal point)
12	blank
13-80	unique name or label for this estimate.

Output:

I/O unit 6: messages.

The name or label of each estimate is written on this unit when the parameter card is read. The number -10.0 is written at the completion of processing for that estimate.

I/O unit 8: density estimates.

The following types of records in the sequence:

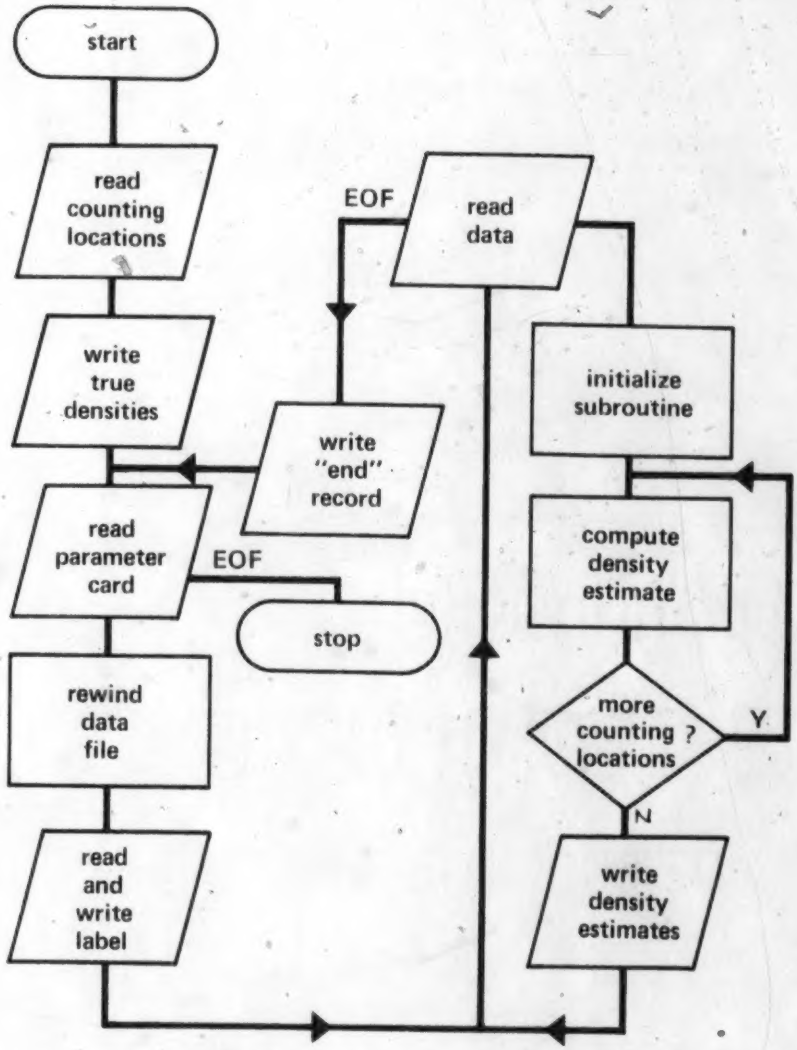
(1) the number of counting locations and the true density at each counting location in the format (I4, 12F10.6).

(2) name or label from one parameter card.

(3) the density estimates for one sample at the several counting locations in the format (12F10.4). Record (3) is repeated for each sample.

(4) the number -10.0 in the format (F10.4). Records (2) through (4) are repeated for each parameter card.

```
C  DENSITY ESTIMATOR TESTING PROGRAM                              DSEM  10
      REAL A(20),B(20),G(20),TITLE(17),S(20)                      DSEM  20
      REAL X(1000),Y(1000),Z(1000),ID(30)                         DSEM  30
      REAL END/-10.0/,DE(20)                                      DSEM  40
C  READ COUNTING LOCATIONS AND WRITE NLOC                         DSEM  50
      READ(4,800)NLOC                                             DSEM  60
      DO 100 I=1,NLOC                                             DSEM  70
  100 READ(4,801)K,A(K),B(K),G(K),S(K)                            DSEM  80
      WRITE(8,802)NLOC,(S(I),I=1,NLOC)                            DSEM  90
                                                                  DSEM 100
C  READ CONTROL CARD, REWIND DATA FILE AND WRITE TITLE            DSEM 110
                                                                  DSEM 120
  200 READ(5,804,END=999)KM,PAR,TITLE                             DSEM 130
      REWIND 1                                                    DSEM 140
      WRITE(8,805)TITLE                                           DSEM 150
      WRITE(6,807)TITLE                                           DSEM 160
C  READ DATA                                                      DSEM 170
  300 READ(1,END=900)N,ID,L,(X(I),Y(I),Z(I),I=1,N)                DSEM 180
C  INITIALIZE SUBROUTINE                                          DSEM 190
      GO TO (310,320,330,340),KM                                  DSEM 200
  310 CONTINUE                                                    DSEM 210
      CALL DESQ(X,Y,Z,N,PAR)                                      DSEM 220
      GO TO 400                                                   DSEM 230
  320 CONTINUE                                                    DSEM 240
      CALL DESKQ(X,Y,Z,N)                                         DSEM 250
      GO TO 400                                                   DSEM 260
  330 CONTINUE                                                    DSEM 270
      CALL DESRQ(X,Y,Z,N,PAR)                                     DSEM 280
      GO TO 400                                                   DSEM 290
  340 CONTINUE                                                    DSEM 300
      CALL DEFQ(X,Y,Z,N,PAR)                                      DSEM 310
C  LOOP THROUGH COUNTING LOCATIONS                                DSEM 320
  400 DO 500 ILOC=1,NLOC                                          DSEM 330
      AI=A(ILOC)                                                  DSEM 340
      BI=B(ILOC)                                                  DSEM 350
      GI=G(ILOC)                                                  DSEM 360
      GO TO (410,420,430,440),KM
  410 CALL DES(AI,BI,GI,D)
      GO TO 490
  420 CALL DESK(AI,BI,GI,D)
      GO TO 490
  430 CALL DESR(AI,BI,GI,D)
      GO TO 490
  440 CALL DEF(AI,BI,GI,D)
  490 DE(ILOC)=D
  500 CONTINUE
C  WRITE DENSITY ESTIMATORS AND RETURN TO READ NEXT SAMPLE
      WRITE(8,806) (DE(I),I=1,NLOC)
      GO TO 300
C  WRITE END-OF-DATA LINE
  900 WRITE(8,806) END
      WRITE(6,806) END
      GO TO 200
  999 STOP
  800 FORMAT(I4)
  801 FORMAT(I3,4F10.6)
  802 FORMAT(I4,12F10.6)
  804 FORMAT(I1,1X,F9.0,1X,17A4)
  805 FORMAT(17A4)
  806 FORMAT(12F10.4)
  807 FORMAT(/1X,17A4)
  808 FORMAT(1X,'SQUARE',F8.3)
  809 FORMAT(1X,'KAMB')
  810 FORMAT(1X,'CONSTANT VARIABILITY',F8.3)
  811 FORMAT(1X,'FISHER',F8.3)
      END
      SUBROUTINE DESQ(X,Y,Z,N,K)
C INITIALIZE CONSTANT-AREA ESTIMATOR
      REAL X(N),Y(N),Z(N),K
      TEST=K/100.0
      TESTN=N*TEST
      RETURN
      ENTRY DES(A,B,G,D)                                                DSEM 730
C COMPUTE ESTIMATE                                                      DSEM 740
      KOUNT=0                                                           DSEM 750
      DO 1 I=1,N                                                        DSEM 760
    1 KOUNT=KOUNT+ABS(A*X(I)+B*Y(I)+G*Z(I))+TEST                        DSEM 770
      D=KOUNT/TESTN                                                     DSEM 780
      RETURN                                                            DSEM 790
      END                                                               DSEM 800
      SUBROUTINE DESKQ(X,Y,Z,N)                                         DSEM 810
C INITIALIZE CONSTANT-AREA ESTIMATOR USING KAMB'S RULE                  DSEM 820
      REAL X(N),Y(N),Z(N)                                               DSEM 830
      TEST=9.0/(N+9.0)                                                  DSEM 840
      TESTN=TEST*N                                                      DSEM 850
      RETURN                                                            DSEM 860
      ENTRY DESK(A,B,G,D)                                               DSEM 870
C COMPUTE DENSITY ESTIMATE                                              DSEM 880
      KOUNT=0                                                           DSEM 890
      DO 1 I=1,N                                                        DSEM 900
    1 KOUNT=KOUNT+ABS(A*X(I)+B*Y(I)+G*Z(I))+TEST                        DSEM 910
      D=KOUNT/TESTN                                                     DSEM 920
      RETURN                                                            DSEM 930
      END                                                               DSEM 940
      SUBROUTINE DESRQ(X,Y,Z,N,V)                                       DSEM 950
C INITIALIZE CONSTANT-PROPORTION ESTIMATOR                              DSEM 960
      REAL X(N),Y(N),Z(N)                                               DSEM 970
      REAL COSR(91),AREA(91)                                            DSEM 980
      INTEGER FREQ(91)                                                  DSEM 990
C  COMPUTE TEST NUMBER AND SET UP COSINES AND AREAS                     DSEM1000
      NTEST=N-N/(V*V*N+1.0)                                             DSEM1010
      DO 100 I=2,90                                                     DSEM1020
      KR=91-I                                                           DSEM1030
      COSR(I)=COS(KR*1.745329E-2)                                       DSEM1040
  100 AREA(I)=1.0-COSR(I)                                               DSEM1050
      COSR(1)=0.0                                                       DSEM1060
      AREA(1)=1.0                                                       DSEM1070
      COSR(91)=2.0                                                      DSEM1080
      AREA(91)=0.0                                                      DSEM1090
      RETURN                                                            DSEM1100
      ENTRY DESR(A,B,G,D)                                               DSEM1110
C COMPUTE DENSITY ESTIMATE                                              DSEM1120
C  ZERO FREQUENCIES                                                     DSEM1130
      DO 200 I=2,91                                                     DSEM1140
  200 FREQ(I)=0                                                         DSEM1150
C  LOOP TO CONSIDER EACH DATA POINT                                     DSEM1160
      DO 300 I=1,N                                                      DSEM1170
      CPHI=ABS(A*X(I)+B*Y(I)+G*Z(I))                                    DSEM1180
      DO 250 M=2,91                                                     DSEM1190
      IF(CPHI.LE.COSR(M))GO TO 300                                      DSEM1200
  250 CONTINUE                                                          DSEM1210
  300 FREQ(M)=FREQ(M)+1                                                 DSEM1220
C  COMPUTE DENSITY ESTIMATE                                             DSEM1230
      NS=0                                                              DSEM1240
      DO 400 M=2,91                                                     DSEM1250
      NS=NS+FREQ(M)                                                     DSEM1260
      IF(NS.GT.NTEST) GO TO 500                                         DSEM1270
  400 CONTINUE                                                          DSEM1280
  500 D=(N-NS+FREQ(M))/(N*AREA(M-1))                                    DSEM1290
      RETURN                                                            DSEM1300
      END                                                               DSEM1310
      SUBROUTINE DEFQ(X,Y,Z,N,CK)                                       DSEM1320
C INITIALIZE FISHER-WEIGHTED ESTIMATOR                                  DSEM1330
      REAL X(N),Y(N),Z(N)                                               DSEM1340
      CONST=CK/N                                                        DSEM1350
      RETURN                                                            DSEM1360
      ENTRY DEF(A,B,G,D)                                                DSEM1370
C COMPUTE DENSITY ESTIMATE                                              DSEM1380
      DENS=0.0                                                          DSEM1390
      DO 1 I=1,N                                                        DSEM1400
    1 DENS=DENS+EXP(CK*(ABS(A*X(I)+B*Y(I)+G*Z(I))-1.0))                 DSEM1410
      D=DENS*CONST                                                      DSEM1420
      RETURN                                                            DSEM1430
      END                                                               DSEM1440
```


start
read counting locations
write true densities
read parameter card
EOF
stop
rewind data file
read and write label
write "end" record
EOF
read data
initialize subroutine
compute density estimate
more counting locations ?
Y
N
write density estimates

Flow Diagram for Program DSEM

Identification

Program name:

DSPDAT1

Program title:

Areal-analysis package - data preparation for display programs.

Author:

John Ramsden

Organization:

Department of Geology, University of Alberta, Edmonton, Alberta, Canada

Date:

1 October 1974

Program version number:

1

Application

Purpose:

To read a data file written by program INPUT1 or INPUTK1 and output four data files suitable for input to programs WNDPLT3, NPPLOT1 and MAP3.

Method:

The program reads a rotation matrix and two parameters. Each block of data read from the input file is written onto output files IOD and IMA. The data is then rotated using the rotation matrix, and the rotated data written onto output files IODR and IMAR. Files IOD and IODR are suitable for input to programs WNDPLT3 and NPPLOT1; files IMA and IMAR are suitable for input to program MAP3.

If the input parameter has a value of 1, the input data are assumed to contain the direction cosines of lines and the trends and plunges of the lines. The data are output as read. If the input parameter has a value of 2, the input data are assumed to contain the direction cosines of the poles to planes and the dip directions and dips of the planes. The dip directions and dips are replaced by the trends and plunges of the poles to the planes before the data are written.

After rotation, new trends and plunges are computed from the direction cosines.

If the output parameter is not zero, all the numbers for display on the map are replaced by zeros. If the output parameter is neither zero nor 1, all the labels to be plotted on the map are replaced by blanks.

Data restrictions:

None.

Timing:

T=1.3+0.0082P where T=CPU time in seconds, P=total number of data points.

Implementation

Original installation:

Organization:	University of Alberta Computing Services.
Computer:	IBM 360 model 67 (1024K).
I/O devices:	card, disc, tape, typewriter terminals and display terminals.
Software:	Michigan Terminal System (MTS), a virtual-storage time-sharing system.

Source language:

FORTRAN IV

Subroutines required:

Subroutines RTSP, SRF from JR subroutine library. Subroutine GMPRD from SSP subroutine library.

Size:

Total size including all required subroutines is 13288 bytes.

Requirements for I/O devices:

I/O unit no.	read/ write	Maximum FORTRAN record length (bytes)	device control statements	suggested device
2	read	30 formatted	none	disc; card.
3	write	1608 unformatted	none	disc; tape.
4	write	1608 unformatted	none	disc; tape.
5	read	3 formatted	none	card; terminal.
6	write	62 formatted.	none	line printer.
7	read	4008 unformatted	none	disc; tape.
8	write	2408 unformatted	none	disc; tape.
9	write	2408 unformatted	none	disc; tape.

Usage

Input:

I/O unit 2: rotation matrix.

Three records, each record containing the corresponding column of the matrix as three real numbers in columns 1-10, 11-20, and 21-30.

I/O unit 5: parameters.

One record containing

col	contents
1	input parameter
3	output parameter

I/O unit 7: input data file.

The input file is as written by programs INPUT1 and INPUTK1. Variables for each observation are: three direction cosines, weight, easting, northing, number to appear on map, label to appear on map, trend or dip direction, plunge or dip.

Output:

I/O unit 3: data file for orientation diagrams.

Variables for each observation are: three direction cosines, weight.

I/O unit 4: data file for orientation diagrams, rotated.

Same as unit 3 output except rotated direction cosines are used.

I/O unit 6: messages.

If the number of variables in the input data file is not 10 an error message is written and the program stops immediately.

I/O unit 8: data file for map.

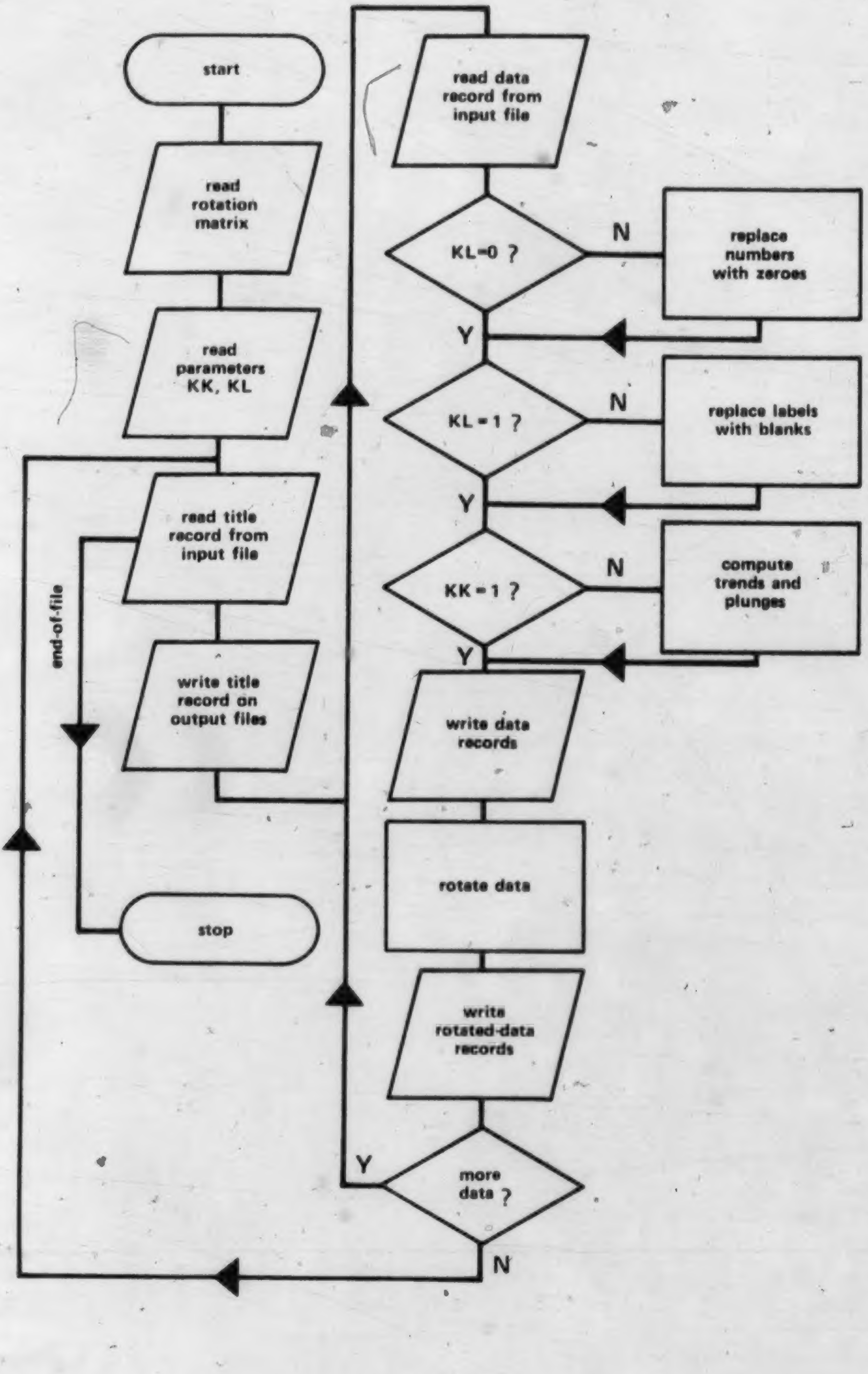
Variables for each observation are: easting, northing, number to appear on map, label to appear on map, trend, plunge.

I/O unit 9: data file for map, rotated.

Same as unit 8 output except rotated trends and plunges are used.

```
C
C
C AREAL-ANALYSIS PACKAGE - PROGRAM "DSPDAT1".
C
C
      INTEGER TITLE(30),DATA(5,100),CT/'T'/,BLANK/'    '/
     *,OV1(5)/'TRUE',' ORI','ENTA','TION','    '/
     *,OV2(5)/'MEAN',' VER','TICA','L   ','    '/
      REAL DCR(3,100),DC(3,100),W(100),T(100),P(100),RM(9),POLR(100)
C
C
    1 FORMAT(15A4)
    2 FORMAT(I10)
    3 FORMAT(A1,30A4)
    4 FORMAT(/1X,'***** ERROR IN KEY FILE *****')
    5 FORMAT(/1X,'***** ERROR - END OF DATA FILE *****')
    6 FORMAT(3F10.6)
    8 FORMAT(1X,30A4)
    9 FORMAT(1X,I3)
   10 FORMAT(I1,1X,I1)
   11 FORMAT(' TOTAL NUMBER OF POINTS=',I5)
   12 FORMAT(/1X,30A4/1X,I6,' PHYSICAL RECORDS; ',I6,' VARIABLES, ',I6,
     *' OBSERVATIONS')
   13 FORMAT(/' ***** ERROR - NUMBER OF VARIABLES IN DATA MATRIX NOT 10
     ******')
   14 FORMAT(1X,9E13.4)
C
C     READ PARAMETERS
C
      IOD=3
      IODR=4
      IMA=8
      IMAR=9
      IOMATR=2
      IOPAR=5
      IOMSG=6
      INPUT=7
C
      READ(IOMATR,6) RM
      READ(IOPAR,10) KK,KL
C
C     READ AND WRITE TITLE RECORD
C
  140 READ(INPUT,END=500) NN,NV,NO,NPHR,TITLE
      NTOT=NO
      WRITE(IOMSG,12) TITLE,NPHR,NV,NO
      IF(NV.NE.10) GOTO910
      NVOD=4
      NNOD=4*NTOT
      NVMA=6
      NNMA=6*NTOT
      DO 410 I=1,5
  410 TITLE(15+I)=OV1(I)
      WRITE(IOD) NNOD,NVOD,NTOT,NPHR,TITLE
      WRITE(IMA) NNMA,NVMA,NTOT,NPHR,TITLE
      DO 412 I=1,5
  412 TITLE(15+I)=OV2(I)
      WRITE(IODR) NNOD,NVOD,NTOT,NPHR,TITLE
      WRITE(IMAR) NNMA,NVMA,NTOT,NPHR,TITLE
```

```
C                                                                       DSPD 600
C     READ AND WRITE DATA RECORDS                                       DSPD 610
C                                                                       DSPD 620
      DO 480 JR=1,NPHR                                                  DSPD 630
      READ(INPUT)NR,IE,((DC(I,J),I=1,3),W(J),(DATA(I,J),I=1,4),         DSPD 640
     *T(J),P(J),J=1,NR)                                                 DSPD 650
      IF(KL.EQ.0)GOTO440                                                DSPD 660
      DO 430 J=1,NR                                                     DSPD 670
  430 DATA(3,J)=0                                                       DSPD 680
      IF(KL.EQ.1)GOTO440                                                DSPD 690
      DO 435 J=1,NR                                                     DSPD 700
  435 DATA(4,J)=BLANK                                                   DSPD 710
  440 IF(KK.EQ.1)GOTO460                                                DSPD 720
      DO 450 J=1,NR                                                     DSPD 730
      T(J)=T(J)+180                                                     DSPD 740
      IF(T(J).GT.360)T(J)=T(J)-360                                      DSPD 750
  450 P(J)=90-P(J)                                                      DSPD 760
  460 WRITE(IOD)NR,IE,((DC(I,J),I=1,3),W(J),J=1,NR)                     DSPD 770
      WRITE(IMA)NR,IE,((DATA(I,J),I=1,4),T(J),P(J),J=1,NR)              DSPD 780
      CALL GMPRD(RM,DC,DCR,3,3,NR)                                      DSPD 790
      CALL RTSP(DCR(1,1),DCR(2,1),DCR(3,1),3,T,P,POLR,1,-1,NR)          DSPD 800
      CALL SRF(T,T,NR,0,999)                                            DSPD 810
      CALL SRF(P,P,NR,0,999)                                            DSPD 820
      WRITE(IODR)NR,IE,((DCR(I,J),I=1,3),W(J),J=1,NR)                   DSPD 830
      WRITE(IMAR)NR,IE,((DATA(I,J),I=1,4),T(J),P(J),J=1,NR)             DSPD 840
  480 CONTINUE                                                          DSPD 850
C                                                                       DSPD 860
  500 STOP                                                              DSPD 870
C                                                                       DSPD 880
  910 WRITE(IOMSG,13)                                                   DSPD 890
      STOP12                                                            DSPD 900
      END                                                               DSPD 910
```


start
read rotation matrix
read parameters KK, KL
read title record from input file
end-of-file
write title record on output files
stop
read data record from input file
KL=0 ?
N
replace numbers with zeroes
Y
KL=1 ?
N
replace labels with blanks
Y
KK=1 ?
N
compute trends and plunges
Y
write data records
rotate data
write rotated-data records
more data ?
Y
N

FLOW DIAGRAM FOR PROGRAM DSPDAT1

Identification

Program name:

GRPKEY1

Program title:

Areal-analysis package - generation of group keys.

Author:

John Ramsden

Organization:

Department of Geology, University of Alberta, Edmonton, Alberta, Canada

Date:

1 October 1974

Program version number:

1

Application

Purpose:

To generate group keys from a traverse file containing data file line numbers.

Method:

From the traverse file the program reads the range of data-file line numbers for each traverse. It then reads group definitions, one at a time. Each group is defined by traverse numbers and observations numbers on each traverse. Traverses specified for a group are processed in order of increasing traverse number, so that line numbers in the key will be in sequence. (This traverse number is not the field traverse number, but a sequence number assigned by the program TRVLST2 so that traverses are numbered sequentially as they appear in the data file.) If the entire traverse is not to be included, the range of line numbers to be included in the key is computed. The line numbers in this range are added to the key.

The number of the traverse just processed is replaced by 10000 and the group definition again searched for the smallest traverse number. When all the traverses included in the group have been processed, the key is written out and the next group definition is read.

Data restrictions:

Maximum number of traverses in traverse file is 999. Maximum number of traverses in one group is 10. Maximum number of line numbers in key is 20.

Timing:

The program required 1.3 seconds of CPU time to generate 32 group keys containing a total of 297 points.

Implementation

Original installation:

Organization: University of Alberta Computing Services.
Computer: IBM 360 model 67 (1024K).
I/O devices: card, disc, tape, typewriter terminals and display terminals.
Software: Michigan Terminal System (MTS), a virtual-storage time-sharing system.

Source language:

FORTRAN IV

Subroutines required:

Subroutines INCH WRITEK from JR subroutine library.

Size:

Total size including all required subroutines is 7768 bytes.

Requirements for I/O devices:

I/O unit no.	read/ write	Maximum FORTRAN record length (bytes)	device control statements	suggested device
1	read	121 formatted	none	disc.
2	read	120 formatted	none	disc.
6	write	121 formatted	none	line printer.
7	write	80 formatted	none	disc; card punch.

Usage

Input:

I/O unit 1; traverse numbers and line-number ranges.

One record for each traverse containing

Col	Contents
2-5	traverse number
13-16	starting line number for traverse
17-20	ending line number for traverse

I/O unit 2: group definitions.

One record for each group containing 120 characters divided into 30 contiguous 4-character fields. From left to right, these fields contain the traverse number, number of first observation to be included, number of last observation to be included. This sequence is repeated for up to 10 traverses. For example, the first six observations on traverse 5 would appear as
...5...1...6
where "." represents a blank. If the number of the first observation to be included is blank or zero, the first observation on the traverse is assumed. If the number of the last observation to be included is blank or zero, the last observation on the traverse is assumed.

Output:

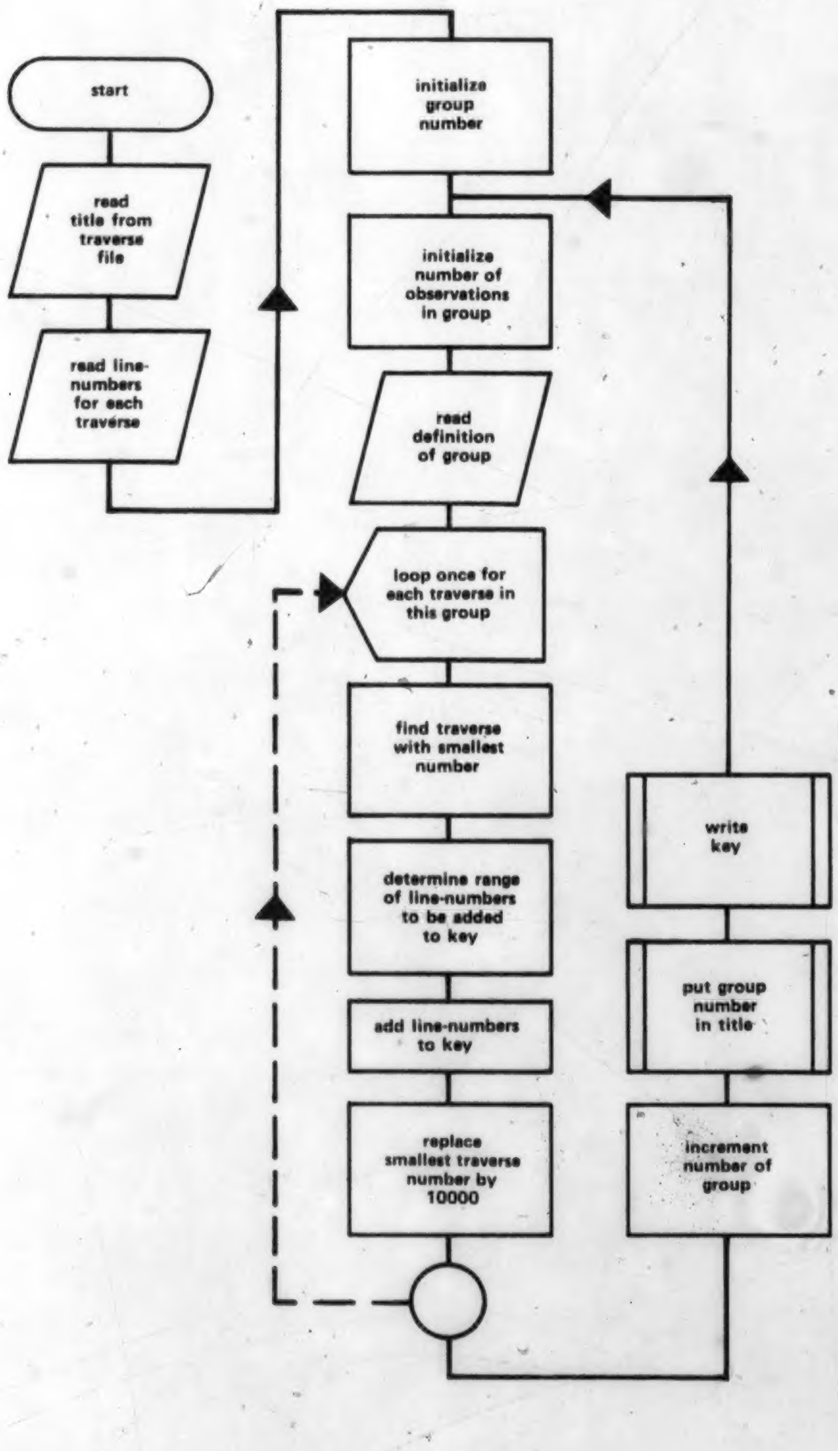
I/O unit 6: error messages.

If the number of observations indicated to be included from one traverse is greater than the number of observations on the traverse, an error message is produced and the program stops immediately.

I/O unit 7: key.

The keys are written by calling the subroutine WRITEK.

```
C
C
C AREAL-ANALYSIS PACKAGE - PROGRAM "GRPKEY1".
C
C
      INTEGER*2 KEY(38),M(999)/999*0/,N(999)/999*0/,L(3,10)
      INTEGER TITLE(30),TITLE2(10)/'GROU','P   ',8*'    '/,FMT(25)
C
    2 FORMAT(30I4)
    3 FORMAT(1X,30A4)
    4 FORMAT(' '/1X,30I4/1X,'TRAVERSE',I3,' ENDING NUMBER TOO LARGE'
     $,10I10)
    5 FORMAT(1X,30I4/1X,10I10)
    6 FORMAT(1X,10I10)
    7 FORMAT(' '/1X,30I4/1X,'TRAVERSE NUMBER',I4,' NOT IN FILE')
    8 FORMAT(2X,25A4)
C
      READ(5,8)FMT
C
      READ(1,3)TITLE
      DO 80 J=1,10
   80 TITLE(J+10)=TITLE2(J)
  100 READ(1,FMT,END=120)J,M(J),N(J)
      LENFIL=N(J)
      GOTO100
  120 CONTINUE
C
      IGROUP=0
  180 CONTINUE
      LENKEY=0
  200 READ(2,2,END=500)L
      NT=10
  240 IF(L(1,NT).NE.0)GOTO250
      NT=NT-1
      GOTO240
  250 CONTINUE
C
      DO 400 I=1,NT
      MIN=10000
      DO 300 J=1,NT
      IF(L(1,J).GE.MIN)GOTO300
      MIN=L(1,J)
      JMIN=J
  300 CONTINUE
      IFIRST=M(MIN)
      ILAST=N(MIN)
      IF(IFIRST.EQ.0 .OR. ILAST.EQ.0)GOTO901
      KA=IFIRST
      KB=ILAST
      IBEGIN=L(2,JMIN)
      IEND=L(3,JMIN)
      IF(IBEGIN.NE.0)KA=IFIRST+IBEGIN-1
      IF(IEND.NE.0)KB=IFIRST+IEND-1
      IF(KB.LE.ILAST)GOTO320
      WRITE(6,4)L,L(1,JMIN),IBEGIN,IEND,KA,KB
      STOP12
  320 CONTINUE
      DO 340 K=KA,KB
      LENKEY=LENKEY+1
      KEY(LENKEY)=K                                                     GRPK 600
  340 CONTINUE                                                          GRPK 610
      L(1,JMIN)=10000                                                   GRPK 620
  400 CONTINUE                                                          GRPK 630
      IGROUP=IGROUP+1                                                   GRPK 640
      CALL INCH(IGROUP,TITLE(13),4,1,' ',IS,IER)                        GRPK 650
      CALL WRITEK(7,TITLE,LENFIL,LENKEY,KEY)                            GRPK 660
      GOTO180                                                           GRPK 670
C                                                                       GRPK 680
  500 STOP                                                              GRPK 690
  901 WRITE(6,7)L,MIN                                                   GRPK 700
      END                                                               GRPK 710
```


start
read title from traverse file
read line-numbers for each traverse
initialize group number
initialize number of observations in group
read definition of group
loop once for each traverse in this group
find traverse with smallest number
determine range of line-numbers to be added to key
add line-numbers to key
replace smallest traverse number by 10000
write key
put group number in title
increment number of group

FLOW DIAGRAM FOR PROGRAM GRPKEY 1

Identification

Program name:

GRPSTA1

Program title:

Areal analysis package - group statistics program.

Author:

John Ramsden

Organization:

Department of Geology, University of Alberta, Edmonton, Alberta, Canada

Date:

1 October 1974

Program version number:

1

Application

Purpose:

To compute statistics for each group of observations for domain analysis.

Method:

The program accepts as parameters the format of the output and the pole of the hemisphere to be used for computing statistics. The data are read from a file written by program INPUTK1. As the data for each group are read, statistics are computed by calling subroutines SCOM1 and FSTAT1. Since these are vector statistics, the pole read from the parameter card is used for assigning sense to the axes.

Data restrictions:

Maximum number of observations in any group is 100.

Timing:

t=0.005p; t=CPU time in seconds; p=total number of points in all groups.

Implementation

Original installation:

Organization: University of Alberta Computing Services.
Computer: IBM 360 model 67 (1024K).
I/O devices: card, disc, tape, typewriter terminals and display terminals.
Software: Michigan Terminal System (MTS), a virtual-storage time-sharing system.

Source language:

FORTRAN IV

Subroutines and function subprograms required:

The following subroutines from JR subroutine library: ANDC FSTAT1 RTSP SCOM1 SRF TAB1 XKF.

Size:

Total size including all required subroutines is 12704 bytes.

Requirements for I/O devices:

I/O unit no.	read/ write	Maximum FORTRAN record length (bytes)	device control statements	suggested device
1	read	2408 unformatted	none	tape; disc.
3	read	30 formatted	none	disc; card.
5	read	102 formatted	none	disc; card.
8	write	determined by given format	none	tape; disc.

Usage

Input:

I/O unit 1: input data file.

As written by program INPUTK1.

I/O unit 3: pole of hemisphere.

Three records containing, in the usual sequence, direction cosines of pole, one direction cosine per record in columns 21-30 with decimal point.

I/O unit 5: output format.

Columns 3-102.

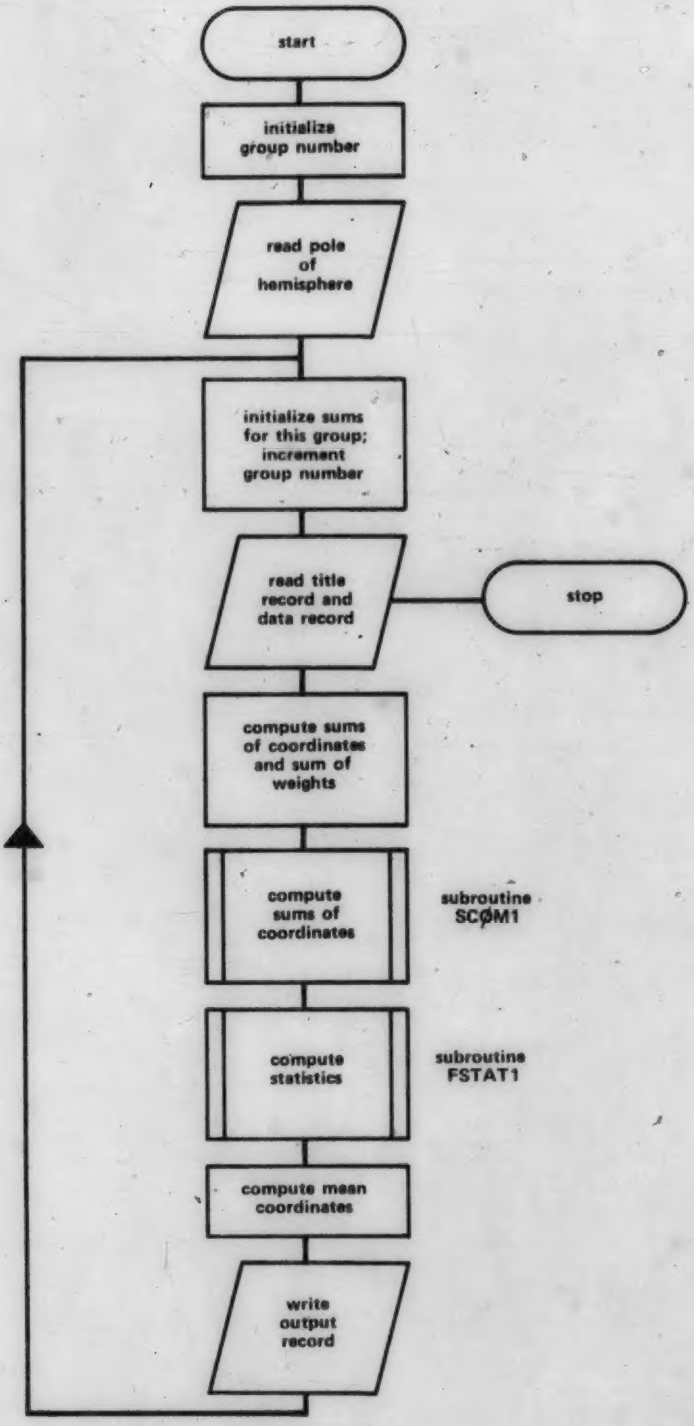
Output:

I/O unit 8: output data file.

Format determined by given format specification. Variables written are: number of group; mean x coordinate of group; mean y coordinate of group; number of observations in group; sum of weights times 10; three sums of components, in the usual sequence, times 10**4; three direction cosines of mean direction, in the usual sequence, times 10**6; length of resultant divided by sum of weights, times 10**6; trend and plunge of mean direction as integers; concentration parameter as integer.

```
C
C
C AREAL-ANALYSIS PACKAGE - PROGRAM "GRPSTA1".
C
C
      INTEGER TITLE(30),IU(3),IE(3),FMT(25)
      INTEGER IX(100),IY(100)
      INTEGER ADDTTL(5)/'GROU','PS  ','    ','    ','    '/
      REAL DC(3,100),E(3),V(3),S(4),U(3),T(2),R(2),DTR/1.745329E-2/
     *,W(100)
      EQUIVALENCE (IU,U),(IE,E),(ISW,SW),(IT,T(2))
      IGRP=0
C
      READ(5,12)FMT
C
C-----READ POLE OF HEMISPHERE
C
      READ(3,5)V
C
C     INITIALIZE VARIABLES FOR THIS GROUP
C
   80 ISX=0
      ISY=0
      SW=0.0
      IGRP=IGRP+1
C
C-----READ DATA AND COMPUTE SUMS OF COORDINATES AND WEIGHTS
C
  100 READ(1,END=400)NN,NV,NO,NPHR,TITLE
      IF(IGRP.NE.1)GOTO110
      DO 105 I=1,5
  105 TITLE(I+10)=ADDTTL(I)
      WRITE(8,11)TITLE
  110 READ(1)N,KE,(IX(J),IY(J),(DC(I,J),I=1,3),W(J),J=1,N)
      DO 115 J=1,NO
  115 WRITE(7,13)J,IX(J),IY(J),(DC(I,J),I=1,3),W(J)
      DO 120 J=1,NO
      ISX=ISX+IX(J)
      ISY=ISY+IY(J)
      SW=SW+W(J)
  120 CONTINUE
C
C-----COMPUTE STATS FOR THIS GROUP
C
  200 N=NO
C     DO 210 J=1,N
C 210 WRITE(6,10)J,(DC(I,J),I=1,3),W(J)
  220 CALL SCOM1(DC,W,N,V,3,U)
C     WRITE(6,14)U
  240 CALL FSTAT1(TITLE,N,SW,U,E,S,T,XK,R,-1)
      ITR=S(1)+0.5
      IPL=S(2)+0.5
C
C-----COMPUTE MEAN COORDINATES
C
  280 IXAVG=FLOAT(ISX)/N+0.5
  285 IYAVG=FLOAT(ISY)/N+0.5
C
C-----WRITE OUTPUT LINE
```

```
C                                                                       GRPS 600
      IF(XK.GT.9999.0) XK=9999.0                                        GRPS 610
      K=XK+0.5                                                          GRPS 620
      CALL SRF(SW,ISW,1,1,999999)                                       GRPS 630
      CALL SRF(U,IU,3,4,9999999)                                        GRPS 640
      CALL SRF(E,IE,3,6,999999)                                         GRPS 650
      CALL SRF(T(2),IT,1,6,999999)                                      GRPS 660
  300 WRITE(8,FMT) IGRP,IXAVG,IYAVG,N,ISW,IU,IE,IT,ITR,IPL,K            GRPS 670
      GOTO80                                                            GRPS 680
C                                                                       GRPS 690
  400 STOP                                                              GRPS 700
C                                                                       GRPS 710
C                                                                       GRPS 720
    1 FORMAT(I1)                                                        GRPS 730
    2 FORMAT(15A4)                                                      GRPS 740
    4 FORMAT(/' **** ERROR IN KEY FILE ****')                           GRPS 750
    5 FORMAT(20X,F10.6)                                                 GRPS 760
    6 FORMAT(/' **** ERROB - END OF DATA FILE ****')                    GRPS 770
    9 FORMAT(A1,30A4)                                                   GRPS 780
   10 FORMAT(1X,I4,1X,4F10.4)                                           GRPS 790
   11 FORMAT('T',30A4)                                                  GRPS 800
   12 FORMAT(2X,25A4)                                                   GRPS 810
   13 FORMAT(1X,I4,2I10,4F10.4)                                         GRPS 820
   14 FORMAT(' SUMS OF COMP ',3F10.6)                                   GRPS 830
      END                                                               GRPS 840
```


start
initialize group number
read pole of hemisphere
initialize sums for this group; increment group number
read title record and data record
stop
compute sums of coordinates and sum of weights
compute sums of coordinates
subroutine SCØM1
compute statistics
subroutine FSTAT1
compute mean coordinates
write output record

FLOW DIAGRAM FOR PROGRAM GRPSTA1

Identification

Program name:

HOMO1

Program title:

Areal-analysis package - fabric homogeneity testing program.

Author:

John Ramsden

Organization:

Department of Geology, University of Alberta, Edmonton, Alberta, Canada

Date:

1 October 1974

Program version number:

1

Application

Purpose:

To test grouped orientation data for homogeneity of concentration parameters and homogeneity of group means interactively.

Method:

The program first reads the group data, then accepts commands from the input device attached to I/O unit 5, normally an interactive terminal but possibly a card reader. In response to these commands the program will add groups to or delete groups from the set of "currently included" groups, and test for homogeneity of concentration parameters or for homogeneity of mean directions among the currently included groups. Details of the commands and their effects follow.

Name: Delete.
Form: "D" followed by optional list of group numbers.
Effect: If group numbers are given, those group numbers are deleted from the list of currently included groups. If no group numbers are given, all groups are deleted from the list.

Name: Restore.
Form: "R" followed by optional list of group numbers.
Effect: If group numbers are given, those group numbers are added to the list of currently included groups. If no group numbers are given, all groups are added to the list.

Name: List.
Form: "L".
Effect: Those group numbers in the list of currently included groups are printed. Those group numbers not in the list are printed separately.

Name: Concentration Parameter Test.
Form: "K".
Effect: Concentration parameters of currently included groups are tested for homogeneity by the subroutine TCPF and the results of the test are printed.

Name: Mean Direction Test.
Form: "M".
Effect: Mean directions of currently included groups are tested for homogeneity by the subroutine TMDF and the results of the test are printed.

Name: Automatic Deletion of Groups until Means are Homogeneous.
Form: "A".
Effect: The group whose mean direction is farthest away from the overall mean direction is deleted from the list of currently included groups. The mean directions of the groups then remaining in the list are tested for homogeneity. If the test indicates that the means are homogeneous a message to that effect is printed and the command "List" is executed before returning to command mode. If the test indicates that the means are not homogeneous the procedure is repeated, a new overall mean direction being computed first. This iteration is continued until either the means become homogeneous or only two groups remain.

Name: Stop.
Form: "S".
Effect: The program requests and reads a name for the set of currently included groups (or "domain"). This set of group numbers is then written on I/O unit 7 and the program stops.

The tests for homogeneity of concentration parameters and homogeneity of group mean directions are taken from Mardia, K. V., 1972, Statistics of Directional Data, London & New York, Academic Press, pp267-271, with the following change: The number of degrees of freedom for the Chi-square statistic used for testing the concentration parameters has been taken as n-1, where n is the number of groups tested, instead of 2n-2 as given by Mardia. Checks performed using data generated artificially from a Fisher distribution with k=50 indicate that the correct number of degrees of freedom is approximately n-1.

Data restrictions:

Maximum number of groups is 100.

Timing:

The program required about 0.5 second of CPU time to read data for 32 groups and test concentration parameters once and test mean directions once.

Implementation

Original installation:

Organization:	University of Alberta Computing Services.
Computer:	IBM 360 model 67 (1024K).
I/O devices:	card, disc, tape, typewriter terminals and display terminals.
Software:	Michigan Terminal System (MTS), a virtual-storage time-sharing system.

Source language:

FORTRAN IV

Subroutines and function subprograms required:

The following subroutines from JR subroutine library: READK WRITEK. The following subroutines from IMSL subroutine library: MDCH MDFD UERTST.

Size:

Total program size including all required library subroutines is 24384 bytes.

Requirements for I/O devices:

I/O unit no.	read/ write	Maximum FORTRAN record length (bytes)	device control statements	suggested device
1	read	3608 unformatted	none	tape; disc.
2	read	80 formatted	none	disc; card.
5	read	61 formatted	none	terminal.
6	write	121 formatted	none	terminal.
7	write	80 formatted	none	disc; card.

Usage

Input:

I/O unit 1: Group data.

As written by program INPUT1 or INPUTK1.

I/O unit 2: Predefined set of group numbers (optional).

If this input is supplied the format must be as written by subroutine WRITEK. If this input is not supplied an end-of-file condition should result when the program attempts to read it.

I/O unit 5: Commands.

Format as follows:

columns	contents
1	code letter
2-5	first group number
6-9	second group number
- - -	- - - -
78-81	twentieth group number

Output:

I/O unit 6: Messages and test results.

I/O unit 7: Final set of group numbers.

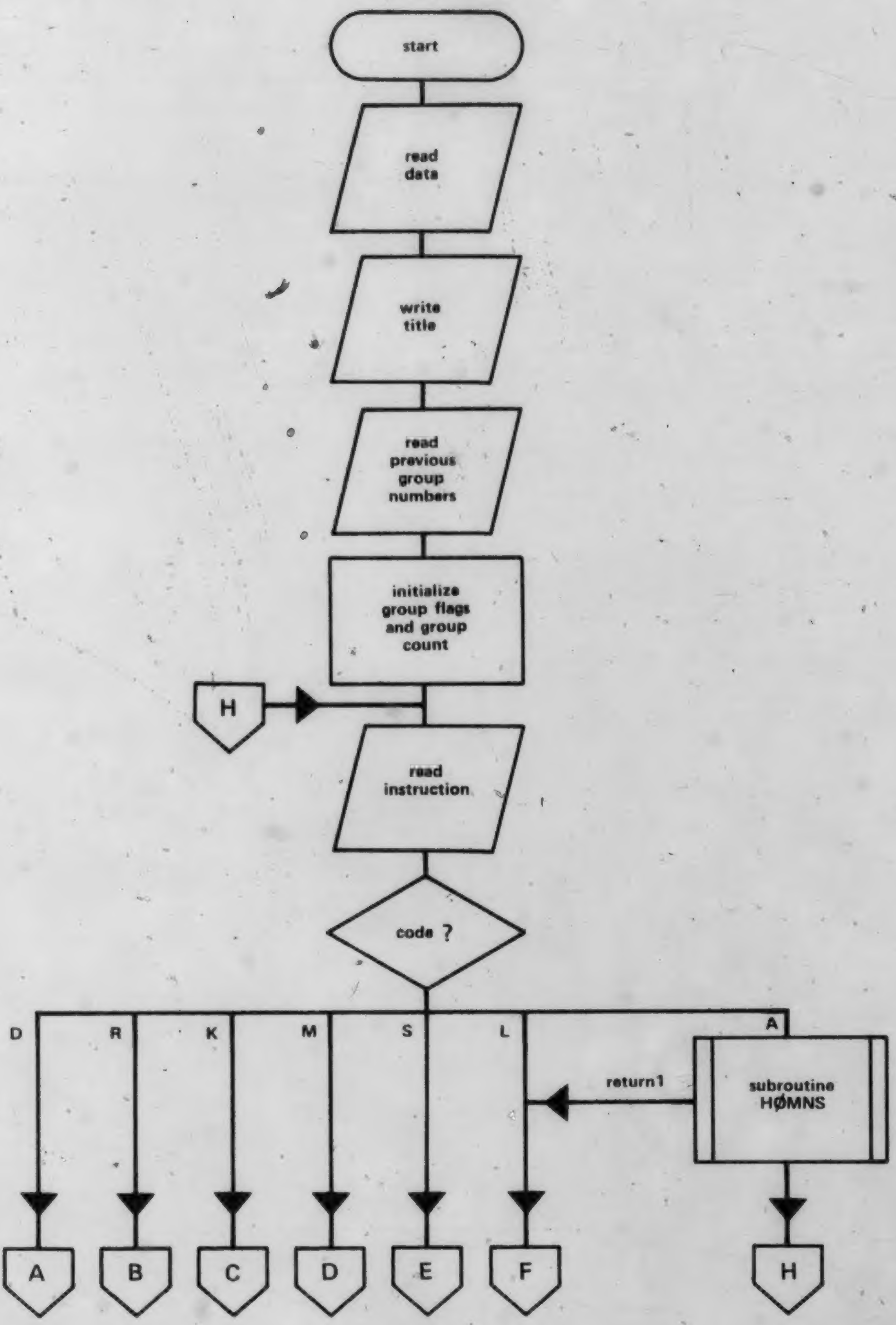
Format as written by subroutine WRITEK.

```
C
C
C AREAL-ANALYSIS PACKAGE - PROGRAM "HOMO1".
C
C
      REAL TITLE(30),R(100),X(100),Y(100),Z(100),T(100),TITLE2(10)
      INTEGER N(100),IS(100),M(100),IG(20)
      INTEGER LETTER(20)/'D','R','K','M','S','L','A'/
      EQUIVALENCE (K,NG,NO)
      INTEGER*2 KEY(118),QU/'&?'/
      COMMON/HOMO1/N,T,R,X,Y,Z,IS,NG
      REAL A(100),B(100),G(100)
C
C
    1 FORMAT('1',30A4/' '/' TEST FOR HOMOGENEITY OF FABRIC'/
     *' ENTER '/
     *'    D TO DELETE GROUPS'/
     *'    R TO RESTORE GROUPS'/
     *'    L TO LIST GROUPS INCLUDED AND DELETED'/
     *'    K TO TEST CONCENTRATION PARAMETERS'/
     *'    M TO TEST MEAN DIRECTIONS'/
     *'    A FOR AUTOMATIC DELETION OF GROUPS UNTIL MEANS ARE HOMOGENEOUS
     *'/
     *'    S TO STOP'/
     *' ')
    2 FORMAT(1X,'ALL GROUPS RESTORED')
    3 FORMAT(1X,'GROUP',I4,' IS ALREADY DELETED')
    4 FORMAT(1X,'GROUP',I4,' IS NOT DELETED')
    5 FORMAT(' DELETED GROUPS ARE')
    7 FORMAT(1X,20I4)
    8 FORMAT(1X,'CHI-SQUARE=',F8.2,10X,'DF=',I4)
    9 FORMAT(' GROUP NUMBER',I4,' IS NOT VALID')
    6 FORMAT(A1,20I3)
   11 FORMAT(18X,I3,F7.1,3F7.1,27X,F9.6)
   12 FORMAT(1X,30A4)
   13 FORMAT(' NONE')
   14 FORMAT(' GROUP',I4,' DELETED')
   15 FORMAT(' GROUP',I4,' RESTORED')
   16 FORMAT(' TAIL PROBABILITY =',F6.3/' '/' ')
   17 FORMAT(' PROBABILITY COULD NOT BE COMPUTED'/' '/' ')
   18 FORMAT(' INCLUDED GROUPS ARE')
   19 FORMAT(' F STATISTIC=',F8.2,10X,'DF=',I4,',',I4)
   20 FORMAT(2I10,4F10.4,I10)
C  21 FORMAT(1X,I4,I10,5F10.6)
   22 FORMAT(1X,'THE FOLLOWING PREVIOUSLY DEFINED DOMAIN HAS BEEN SET'/
     *1X,30A4)
   23 FORMAT(A2)
   24 FORMAT(1X,'ENTER NAME OF DOMAIN'/'&?')
   25 FORMAT(10A4)
   26 FORMAT(//' ***** FATAL ERROR - NUMBER OF GROUPS IN INPUT >100. NO
     *CAN DO. SORRY! ')
C
C
      READ(1)NN,NV,NO,NPHR,TITLE
      IF(NO.GT.100)GOTO4000
      READ(1)NO,IE,(N(J),T(J),X(J),Y(J),Z(J),A(J),B(J),G(J),R(J),J=1,NO)
      K=NO
      WRITE(6,1)TITLE
C
```

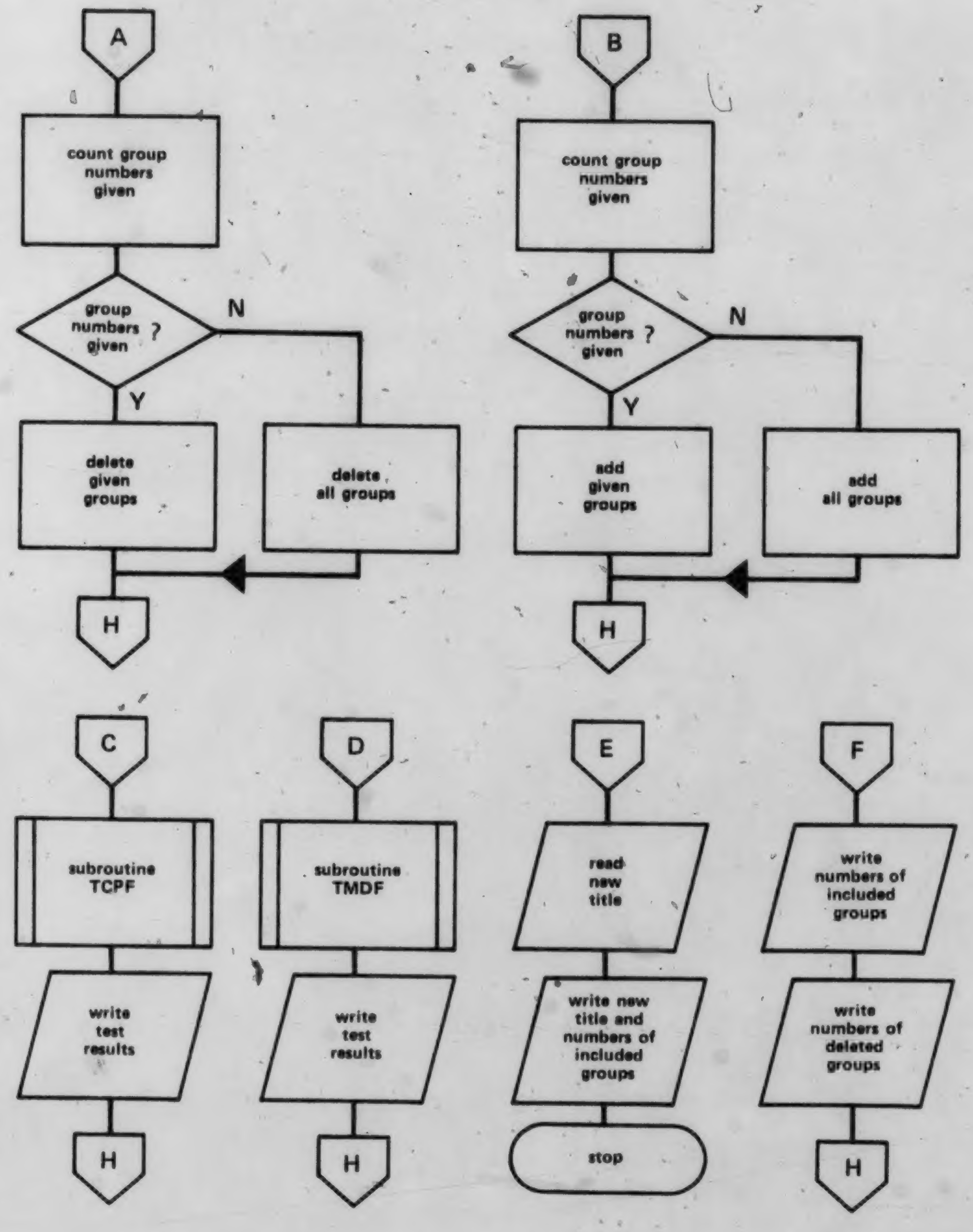
```
C     CHECK FILE FOR PREVIOUSLY DEFINED DOMAIN                          HOMO 600
C                                                                       HOMO 610
      CALL READK(2,TITLE,LENFIL,LENK,KEY,&110,&110)                     HOMO 620
      DO 60 I=1,K                                                       HOMO 630
   60 IS(I)=0                                                           HOMO 640
      IKEY=0                                                            HOMO 650
   65 IKEY=IKEY+1                                                       HOMO 660
      IF(IKEY.GT.LENK)GOTO70                                            HOMO 670
      I=KEY(IKEY)                                                       HOMO 680
      IS(I)=1                                                           HOMO 690
      GOTO65                                                            HOMO 700
   70 WRITE(6,22)TITLE                                                  HOMO 710
      GOTO700                                                           HOMO 720
C                                                                       HOMO 730
C     SET ALL GROUPS "ON"                                               HOMO 740
C                                                                       HOMO 750
  110 DO 115 I=1,K                                                      HOMO 760
  115 IS(I)=1                                                           HOMO 770
C                                                                       HOMO 780
C     WRITE "?" AND READ NEXT COMMAND                                   HOMO 790
C                                                                       HOMO 800
  120 WRITE(6,23)QU                                                     HOMO 810
      READ(5,6)L,IG                                                     HOMO 820
      DO 140 I=1,20                                                     HOMO 830
      IF(L.[illegible]TTER(I))GOTO160                                   HOMO 840
  140 CONTINUE                                                          HOMO 850
      GOTO120                                                           HOMO 860
  160 GOTO(200,300,400,500,600,700,800,900,1000,1100,1200,1300,1400,    HOMO 870
     *1500,1600,1700,1800,1900,2000,2100),I                             HOMO 880
C                                                                       HOMO 890
C     DELETE GROUPS                                                     HOMO 900
C                                                                       HOMO 910
  200 J=20                                                              HOMO 920
      DO 220 I=1,20                                                     HOMO 930
      IF(IG(J).NE.0)GOTO240                                             HOMO 940
      J=J-1                                                             HOMO 950
  220 CONTINUE                                                          HOMO 960
  240 IF(J.EQ.0)GOTO280                                                 HOMO 970
      DO 250 I=1,J                                                      HOMO 980
  250 IS(IG(I))=0                                                       HOMO 990
      GOTO120                                                           HOMO1000
  280 DO 290 I=1,NG                                                     HOMO1010
  290 IS(I)=0                                                           HOMO1020
      GOTO120                                                           HOMO1030
C                                                                       HOMO1040
C     RESTORE GROUPS                                                    HOMO1050
C                                                                       HOMO1060
  300 J=20                                                              HOMO1070
      DO 320 I=1,20                                                     HOMO1080
      IF(IG(J).NE.0)GOTO340                                             HOMO1090
      J=J-1                                                             HOMO1100
  320 CONTINUE                                                          HOMO1110
  340 IF(J.EQ.0)GOTO380                                                 HOMO1120
      DO 350 I=1,J                                                      HOMO1130
  350 IS(IG(I))=1                                                       HOMO1140
      GOTO120                                                           HOMO1150
  380 DO 390 I=1,NG                                                     HOMO1160
  390 IS(I)=1                                                           HOMO1170
      GOTO120                                                           HOMO1180
C                                                                       HOMO1190
```

```
C      TEST CONCENTRATION PARAMETERS
C
  400 CONTINUE
      CALL TCPF(U,NDF)
      WRITE(6,8) U,NDF
      DF=NDF
      CALL MDCH(U,DF,P,IER)
      P=1.0-P
      IF(IER.NE.0)GOTO450
      WRITE(6,16) P
      GOTO120
  450 WRITE(6,17)
      GOTO120
C
C      TEST MEANS
C
  500 CONTINUE
  510 CALL TMDF(F,ND1,ND2)
      WRITE(6,19) F,ND1,ND2
      CALL MDFD(F,ND1,ND2,P,IER)
      IF(IER.NE.0)GOTO120
      P=1.0-P
      WRITE(6,16) P
      GOTO120
C
C      PRINT LIST OF GROUP NUMBERS
C
  700 CONTINUE
      L=0
      DO 720 I=1,NG
      IF(IS(I).EQ.0)GOTO720
      L=L+1
      M(L)=I
  720 CONTINUE
      WRITE(6,18)
      IF(L.EQ.0) WRITE(6,13)
      IF(L.NE.0) WRITE(6,7) (M(I),I=1,L)
      L=0
      DO 740 I=1,NG
      IF(IS(I).EQ.1)GOTO740
      L=L+1
      M(L)=I
  740 CONTINUE
      WRITE(6,5)
      IF(L.EQ.0) WRITE(6,13)
      IF(L.NE.0) WRITE(6,7) (M(I),I=1,L)
      GOTO120
C
C
  600 CONTINUE
      LENK=0
      DO 620 J=1,NO
      IF(IS(J).EQ.0)GOTO620
      LENK=LENK+1
      KEY(LENK)=J
  620 CONTINUE
      WRITE(6,24)
      READ(5,25)TITLE2
      DO 640 I=1,10
  640 TITLE(16+I)=TITLE2(I)
```

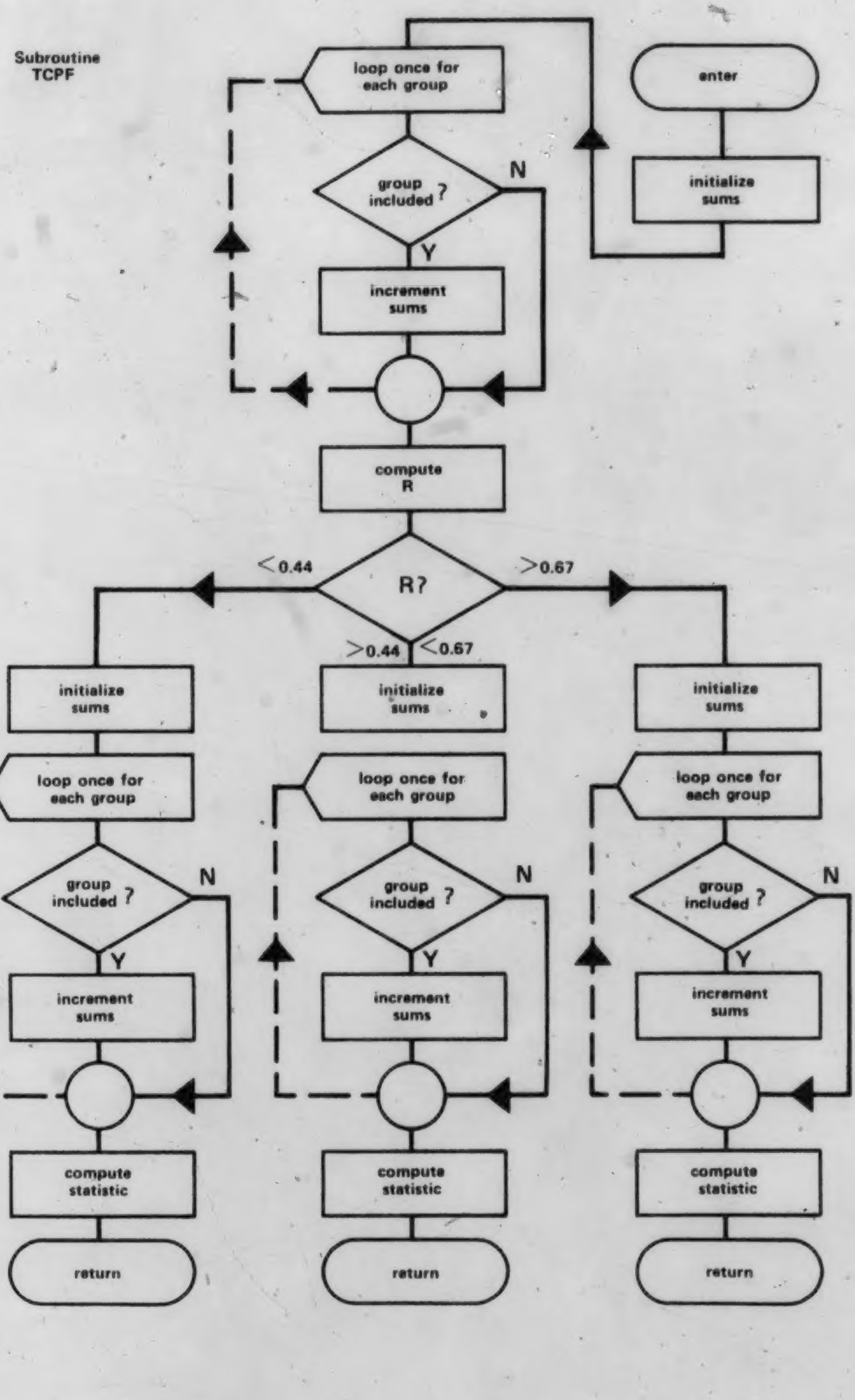
```
      CALL WRITEK(7,TITLE,NO,LENK,KEY)                                  HOMO1800
      STOP                                                              HOMO1810
C                                                                       HOMO1820
C                                                                       HOMO1830
  800 CONTINUE                                                          HOMO1840
      CALL HOMNS(A,B,G,&700)                                            HOMO1850
      GOTO120                                                           HOMO1860
  900 GOTO120                                                           HOMO1870
 1000 GOTO120                                                           HOMO1880
 1100 GOTO120                                                           HOMO1890
 1200 GOTO120                                                           HOMO1900
 1300 GOTO120                                                           HOMO1910
 1400 GOTO120                                                           HOMO1920
 1500 GOTO120                                                           HOMO1930
 1600 GOTO120                                                           HOMO1940
 1700 GOTO120                                                           HOMO1950
 1800 GOTO120                                                           HOMO1960
 1900 GOTO120                                                           HOMO1970
 2000 GOTO120                                                           HOMO1980
 2100 GOTO120                                                           HOMO1990
 4000 WRITE(6,26)                                                       HOMO2000
      STOP12                                                            HOMO2010
      END                                                               HOMO2020
      SUBROUTINE TCPF(U,NDF)                                            HOMO2030
C COMPUTES THE STATISTIC TO TEST FOR HOMOGENEITY OF CONCENTRATION       HOMO2040
C PARAMETERS (FISHER STATISTICS)                                        HOMO2050
      REAL R(100),X(100),Y(100),Z(100),T(100)                           HOMO2060
      INTEGER N(100),IS(100)                                            HOMO2070
      COMMON/HOMO1/N,T,R,X,Y,Z,IS,K                                     HOMO2080
C                                                                       HOMO2090
C                                                                       HOMO2100
      TT=0.0                                                            HOMO2110
      XX=0.0                                                            HOMO2120
      YY=0.0                                                            HOMO2130
      ZZ=0.0                                                            HOMO2140
      NT=0                                                              HOMO2150
      L=0                                                               HOMO2160
      DO 100 I=1,K                                                      HOMO2170
      IF(IS(I).EQ.0)GOTO100                                             HOMO2180
      TT=TT+T(I)                                                        HOMO2190
      XX=XX+X(I)                                                        HOMO2200
      YY=YY+Y(I)                                                        HOMO2210
      ZZ=ZZ+Z(I)                                                        HOMO2220
      NT=NT+N(I)                                                        HOMO2230
      L=L+1                                                             HOMO2240
  100 CONTINUE                                                          HOMO2250
      RT=SQRT(XX*XX+YY*YY+ZZ*ZZ)                                        HOMO2260
      RBT=RT/TT                                                         HOMO2270
      IF(RBT.LT.0.44)GOTO200                                            HOMO2280
      IF(RBT.GT.0.67)GOTO400                                            HOMO2290
      GOTO300                                                           HOMO2300
C                                                                       HOMO2310
  200 SW=0.0                                                            HOMO2320
      SWG=0.0                                                           HOMO2330
      SWGG=0.0                                                          HOMO2340
      DO 220 I=1,K                                                      HOMO2350
      IF(IS(I).EQ.0)GOTO220                                             HOMO2360
      W=5.0*(N(I)-5)/3.0                                                HOMO2370
      G=ARSIN((3.0*R(I))/2.2361)                                        HOMO2380
      WG=W*G                                                            HOMO2390
      SW=SW+W                                                           HOMO2400
      SWG=SWG+WG                                                        HOMO2410
      SWGG=SWGG+WG*G                                                    HOMO2420
  220 CONTINUE                                                          HOMO2430
  230 U=SWGG-SWG*SWG/SW                                                 HOMO2440
      NDF=L-1                                                           HOMO2450
      RETURN                                                            HOMO2460
C                                                                       HOMO2470
  300 SW=0.0                                                            HOMO2480
      SWG=0.0                                                           HOMO2490
      SWGG=0.0                                                          HOMO2500
      DO 320 I=1,K                                                      HOMO2510
      IF(IS(I).EQ.0)GOTO320                                             HOMO2520
      W=(N(I)-4)/0.39356                                                HOMO2530
      G=ARSIN((R(I)+0.17595)/1.02903)                                   HOMO2540
      WG=W*G                                                            HOMO2550
      SW=SW+W                                                           HOMO2560
      SWG=SWG+WG                                                        HOMO2570
      SWGG=SWGG+WG*G                                                    HOMO2580
  320 CONTINUE                                                          HOMO2590
  330 U=SWGG-SWG*SWG/SW                                                 HOMO2600
      NDF=L-1                                                           HOMO2610
      RETURN                                                            HOMO2620
C                                                                       HOMO2630
  400 SR=0.0                                                            HOMO2640
      SVL=0.0                                                           HOMO2650
      SVI=0.0                                                           HOMO2660
      DO 420 I=1,K                                                      HOMO2670
      IF(IS(I).EQ.0)GOTO420                                             HOMO2680
      SR=SR+R(I)*N(I)                                                   HOMO2690
      V=2.0*(N(I)-1)                                                    HOMO2700
      VI=1.0/V                                                          HOMO2710
      SVI=SVI+VI                                                        HOMO2720
      VL=V*ALOG(N(I)*(1.0-R(I))/V)                                      HOMO2730
      SVL=SVL+VL                                                        HOMO2740
  420 CONTINUE                                                          HOMO2750
      V=2.0*(NT-L)                                                      HOMO2760
      D=(SVI-1.0/V)/(3.0*(L-1))                                         HOMO2770
  450 U=(V*ALOG((NT-SR)/V)-SVL)/(1.0+D)                                 HOMO2780
      NDF=L-1                                                           HOMO2790
      RETURN                                                            HOMO2800
      END                                                               HOMO2810
      SUBROUTINE TMDF(F,ND1,ND2)                                        HOMO2820
C COMPUTES THE STATISTIC TO TEST FOR HOMOGENEITY OF SAMPLE MEANS        HOMO2830
C (FISHER STATISTICS)                                                   HOMO2840
      REAL R(100),X(100),Y(100),Z(100),T(100)                           HOMO2850
      INTEGER N(100),IS(100)                                            HOMO2860
      COMMON/HOMO1/N,T,R,X,Y,Z,IS,NG                                    HOMO2870
C                                                                       HOMO2880
C                                                                       HOMO2890
      TT=0.0                                                            HOMO2900
      SR=0.0                                                            HOMO2910
      XX=0.0                                                            HOMO2920
      YY=0.0                                                            HOMO2930
      ZZ=0.0                                                            HOMO2940
      NN=0                                                              HOMO2950
      NQ=0                                                              HOMO2960
      DO 100 I=1,NG                                                     HOMO2970
      IF(IS(I).EQ.0)GOTO100                                             HOMO2980
      NN=NN+N(I)                                                        HOMO2990
      NQ=NQ+1
      TT=TT+T(I)
      XX=XX+X(I)
      YY=YY+Y(I)
      ZZ=ZZ+Z(I)
      SR=SR+R(I)*N(I)
  100 CONTINUE
C
  110 RR=SQRT(XX*XX+YY*YY+ZZ*ZZ)
      RR=RR*NN/TT
  120 RRB=RR/NN
  130 IF(RRB.LT.0.32)GOTO901
  150 F=(NN-NQ)*(SR-RR)/((NN-SR)*(NQ-1))
      IF(RRB.GT.0.67)GOTO200
      CALL XKF(RRB,NN,AK,IE)
      IF(IE.NE.0)GOTO902
      F=F*(1.0-1.0/(5.0*AK*AK))
  200 CONTINUE

      ND1=2*NQ-2
      ND2=2*(NN-NQ)
      RETURN
C
  901 WRITE(6,1)RRB
    1 FORMAT(//' **** SUBROUTINE TMDF: R/N=',E11.4,' IS SMALLER THAN AL
     *LOWABLE RANGE FOR THIS SUBROUTINE'//)
      STOP 12
  902 WRITE(6,2)IE
    2 FORMAT(//' **** SUBROUTINE TMDF: RETURN CODE',I3,' FROM SUBROUTINE
     * "XKF"')
      STOP12
      END
      SUBROUTINE HOMNS(A,B,G,*)
C DELETES GROUPS UNTIL THE MEANS ARE HOMOGENEOUS
      REAL A(100),B(100),G(100),X(100),Y(100),Z(100),T(100),R(100)
      INTEGER N(100),IS(100),NG
      COMMON/HOMO1/N,T,R,X,Y,Z,IS,NG
C
    1 FORMAT(' SUBROUTINE "HOMNS": R IS ZERO')
    2 FORMAT(' GROUP',I4,' HAS BEEN DELETED',5X,I4,' GROUPS REMAIN')
    3 FORMAT(' RETURN CODE',I3,' FROM IMSL SUBROUTINE "MDFD"')
    4 FORMAT('HOMOGENEITY OF MEANS COULD NOT BE OBTAINED')
    5 FORMAT(' HOMOGENEITY OF MEANS ACCEPTED')
   16 FORMAT(' TAIL PROBABILITY =',F6.3/' ')
   19 FORMAT(' F STATISTIC=',F8.2,10X,'DF=',I4,',',I4)
C
C
  100 CONTINUE
      SX=0.0
      SY=0.0
      SZ=0.0
      NI=0
      DO 200 J=1,NG
      IF(IS(J).EQ.0)GOTO200
      NI=NI+1
      SX=SX+X(J)
      SY=SY+Y(J)
      SZ=SZ+Z(J)
  200 CONTINUE
      RR=SQRT(SX*SX+SY*SY+SZ*SZ)
      IF(RR.EQ.0) WRITE(6,1)                                            HOMO3600
      ARR=SX/RR                                                         HOMO3610
      BRR=SY/RR                                                         HOMO3620
      GRR=SZ/RR                                                         HOMO3630
      CMIN=1.000                                                        HOMO3640
      DO 300 J=1,NG                                                     HOMO3650
      IF(IS(J).EQ.0) GOTO300                                            HOMO3660
      COS=ARR*A(J)+BRR*B(J)+GRR*G(J)                                    HOMO3670
      IF(COS.GE.CMIN) GOTO300                                           HOMO3680
      CMIN=COS                                                          HOMO3690
      JMIN=J                                                            HOMO3700
  300 CONTINUE                                                          HOMO3710
      IS(JMIN)=0                                                        HOMO3720
      NI=NI-1                                                           HOMO3730
      WRITE(6,2) JMIN,NI                                                HOMO3740
      CALL TMDF(F,ND1,ND2)                                              HOMO3750
      WRITE(6,19) F,ND1,ND2                                             HOMO3760
      CALL MDFD(F,ND1,ND2,P,IER)                                        HOMO3770
      IF(IER) 360,380,360                                               HOMO3780
  360 WRITE(6,3) IER                                                    HOMO3790
      RETURN                                                            HOMO3800
  380 P=1.0-P                                                           HOMO3810
      WRITE(6,16) P                                                     HOMO3820
      IF(P-0.05) 400,500,500                                            HOMO3830
  400 IF(NI-2) 420,420,100                                              HOMO3840
  420 WRITE(6,4)                                                        HOMO3850
      RETURN                                                            HOMO3860
  500 WRITE(6,5)                                                        HOMO3870
      RETURN1                                                           HOMO3880
      END                                                               HOMO3890
```


start
read data
write title
read previous group numbers
initialize group flags and group count
H
read instruction
code ?
D
R
K
M
S
L
A
return1
subroutine HØMNS
A
B
C
D
E
F
H

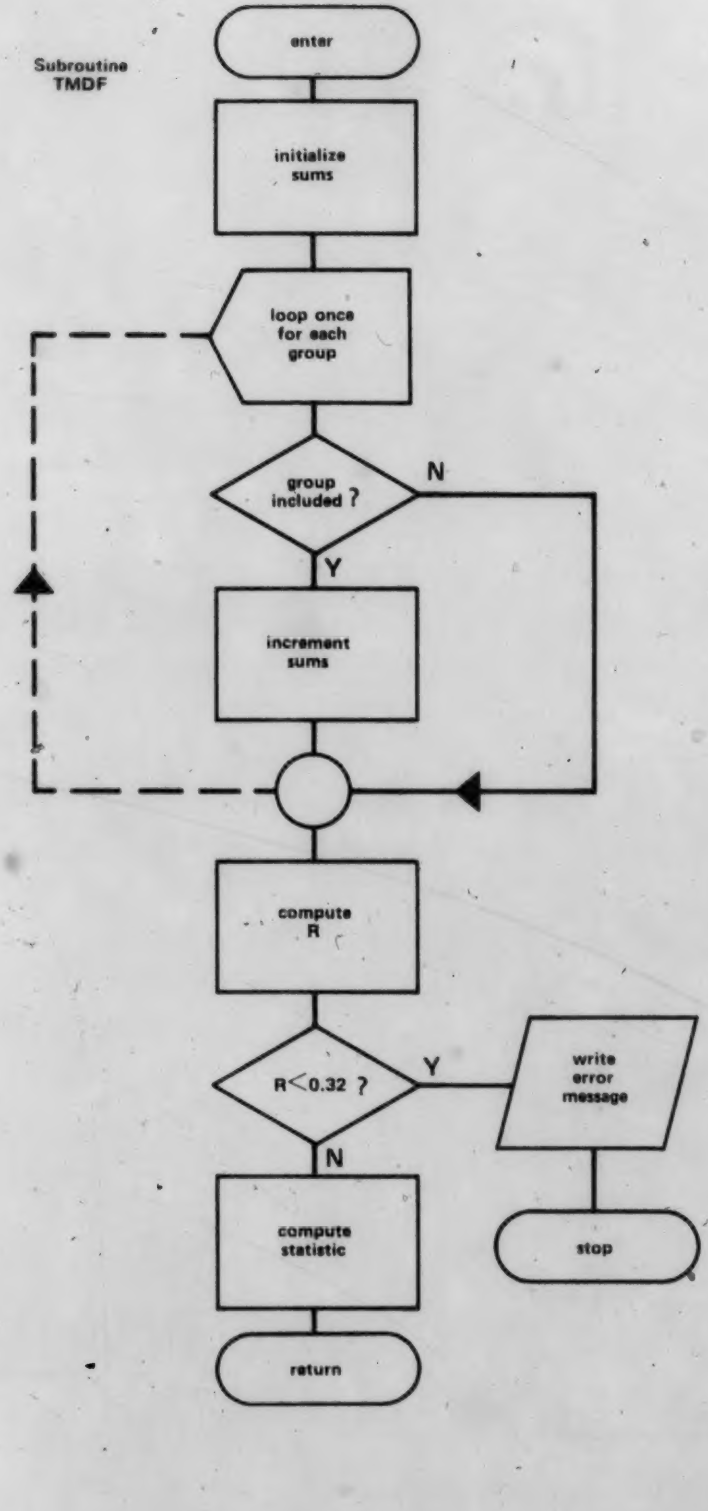
FLOW DIAGRAM FOR PROGRAM HOM01 - PART 1


A
count group numbers given
group numbers given ?
N
Y
delete given groups
delete all groups
H
B
count group numbers given
group numbers given ?
N
Y
add given groups
add all groups
H
C
subroutine TCPF
write test results
H
D
subroutine TMDF
write test results
H
E
read new title
write new title and numbers of included groups
stop
F
write numbers of included groups
write numbers of deleted groups
H

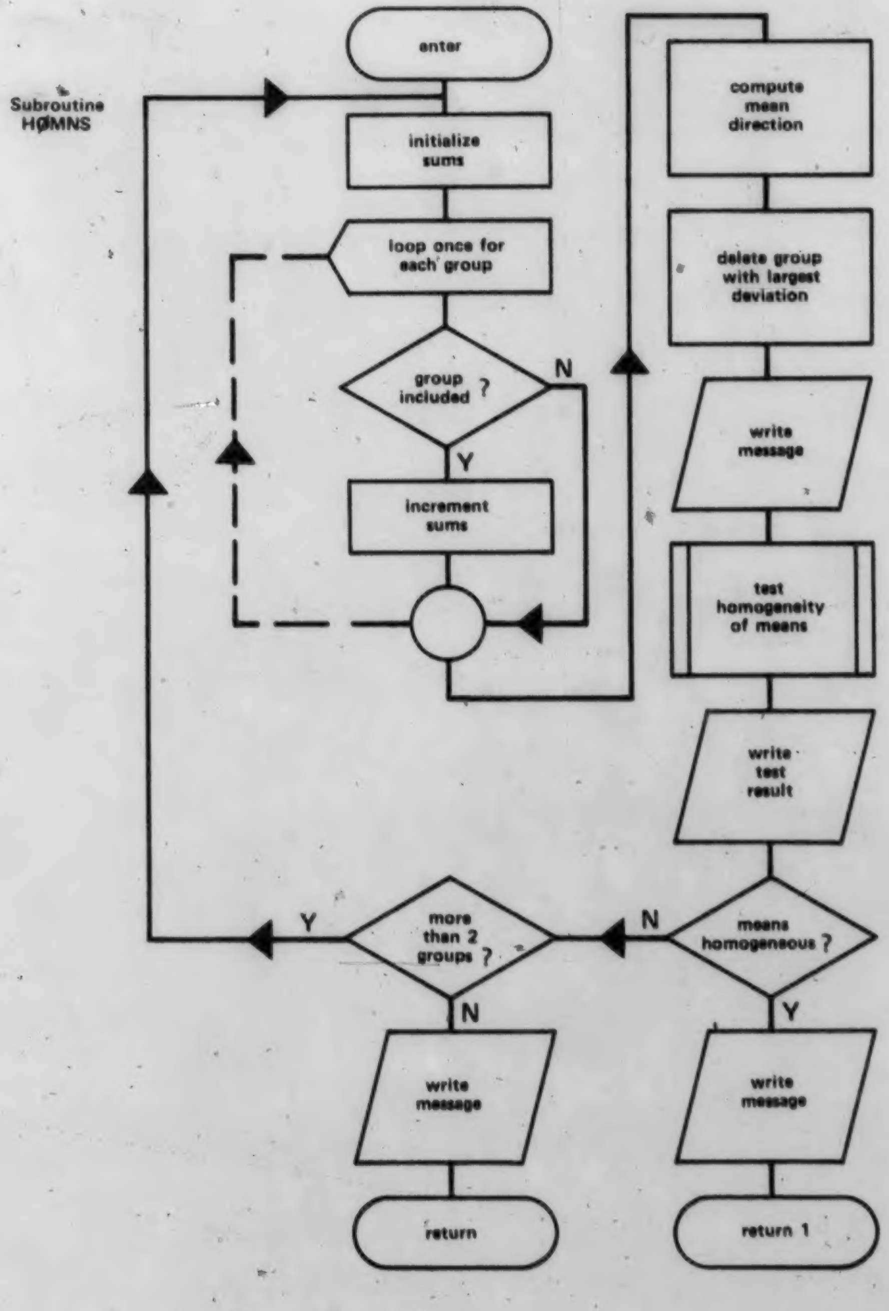
FLOW DIAGRAM FOR PROGRAM HOM01 - PART 2


Subroutine
TCPF
enter
initialize sums
loop once for each group
group included ?
N
Y
increment sums
compute R
R?
<0.44
>0.67
>0.44 <0.67
initialize sums
loop once for each group
group included ?
N
Y
increment sums
compute statistic
return
initialize sums
loop once for each group
group included ?
N
Y
increment sums
compute statistic
return
initialize sums
loop once for each group
group included ?
N
Y
increment sums
compute statistic
return

FLOW DIAGRAM FOR PROGRAM HOMO1 - PART 3


Subroutine TMDF
enter
initialize sums
loop once for each group
group included ?
N
Y
increment sums
compute R
R<0.32 ?
Y
write error message
stop
N
compute statistic
return

FLOW DIAGRAM FOR PROGRAM HOMO1 - PART 4


Subroutine
HØMNS
enter
initialize sums
loop once for each group
group included ?
N
Y
increment sums
compute mean direction
delete group with largest deviation
write message
test homogeneity of means
write test result
means homogeneous ?
N
Y
more than 2 groups ?
Y
N
write message
write message
return
return 1

FLOW DIAGRAM FOR PROGRAM HOMO1 - PART 5

Identification

Program name:

INPUTK1

Program title:

Areal analysis package - data input program (key method).

Author:

John Ramsden

Organization:

Department of Geology, University of Alberta, Edmonton, Alberta, Canada

Date:

1 October 1974

Program version number:

1

Application

Purpose:

To read specified variables from an input file using a key and write the data in a convenient form for use by other programs in the areal-analysis package.

Method:

The program accepts as parameters the number of variables to be read from each record of the input file and the format for reading those variables. The first field in the format specification must refer to a line-type character and must be specified as A1. The specified number of variables must include this field. Since the data array is declared as integer, and some FORTRAN input routines will not allow data to be read into an integer variable with an "F" format code (that is, as a real number), all input data must be in integer form and all fields must be specified with an "I" format code. If it is desired to convert integers to real numbers before they are placed in the output file, an additional set of parameters may be added to the parameter card containing the input format. These parameters specify which input variables are to be converted to real numbers and how many decimal places each should have.

Which records of the input file are read is determined by a "key" of record numbers. At each record of the input file whose number appears in the key the program reads the specified number of fullword variables using the specified format. The data are written unformatted into the output file as a header record followed by data records. Each output data record except the last always contains the data from 100 input records.

The output data records are first written onto a scratch file. After all the input records have been read, the header record is written onto the output file, and the output data records are then copied from the scratch file to the output file.

If the first byte of the first input record contains the letter "T" this record is assumed to be a title record and counting of input records begins with record 2. Otherwise counting of the input records begins with record 1. Since only those input records are read whose numbers appear in the key, an end-of-file condition resulting from an attempt to read the input file always indicates an error, and the program stops after printing a message. After processing the input records as directed by a key, the program attempts to read another key. If an end-of-file condition results the program stops normally. If another key is read the input file is rewound and the process repeated using the new key. In this way any number of sets of data may be extracted from the input file and written sequentially on the output file.

Data restrictions:

Maximum length of the given format is 60 characters. Maximum number of variables read from each input record is 10. Maximum number of input record numbers in one key is 5000.

Timing:

t=0.004pv; t=CPU time in seconds; p=total number of data points; v=number of variables.

Implementation

Original installation:

Organization: University of Alberta Computing Services.
Computer: IBM 360 model 67 (1024K).
I/O devices: card, disc, tape, typewriter terminals and display terminals.
Software: Michigan Terminal System (MTS), a virtual-storage time-sharing system.

Source language:

FORTRAN IV

Subroutines and function subprograms required:

The following subroutine from JR subroutine library: READK.

Size:

Total program size including all required subroutines is 17464 bytes.

Requirements for I/O devices:

I/O unit no.	read/ write	Maximum FORTRAN record length (bytes)	device control statements	suggested device
1	read	determined by given format	rewind	tape; disc; cards.
2	read	80 formatted	none	disc.
5	read	102 formatted	none	cards.
6	write	35 formatted	none	line printer.
7	read, write	8+400*nv, unformatted	rewind	tape; disc.
8	write	8+400*nv, unformatted	none	tape; disc.

nv=number of variables.

Usage

Input:

I/O unit 1: input data file.

All records have the same format unless the first record is a title record, in which case the first record contains the letter "T" in column 1 and a title in columns 2-121.

I/O unit 2: keys.

Format as written by subroutine WRITEK.

I/O unit 5: parameters.

One recording containing

columns	contents
1-2	number of variables to be read
3-62	format
63-102	up to 20 2-digit numbers. These numbers are interpreted in pairs: the first number of each pair is taken as the number of a variable to be converted to a real number, and the second number of each pair is taken as the number of decimal places. For example, the pair of numbers " 4 3" indicates that the fourth variable (not counting the line-type character) is to be divided by 10**3 and the result placed in the output record as a real number.

Output:

I/O unit 6: messages.

The program writes the number of output data records. Error messages are self-explanatory.

I/O unit 7: scratch file.

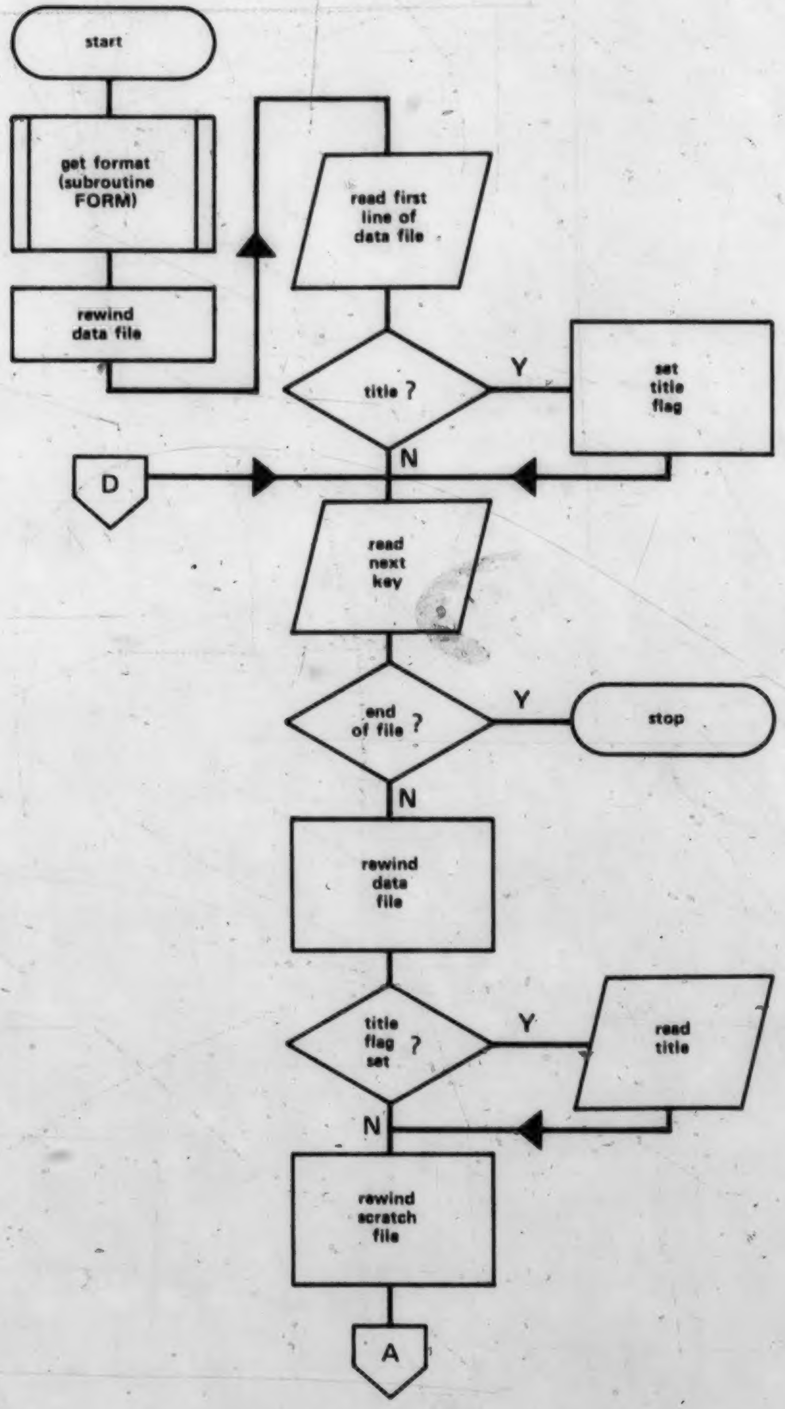
Output data records are stored in this file until they are copied to the output file.

I/O unit 8: output file.

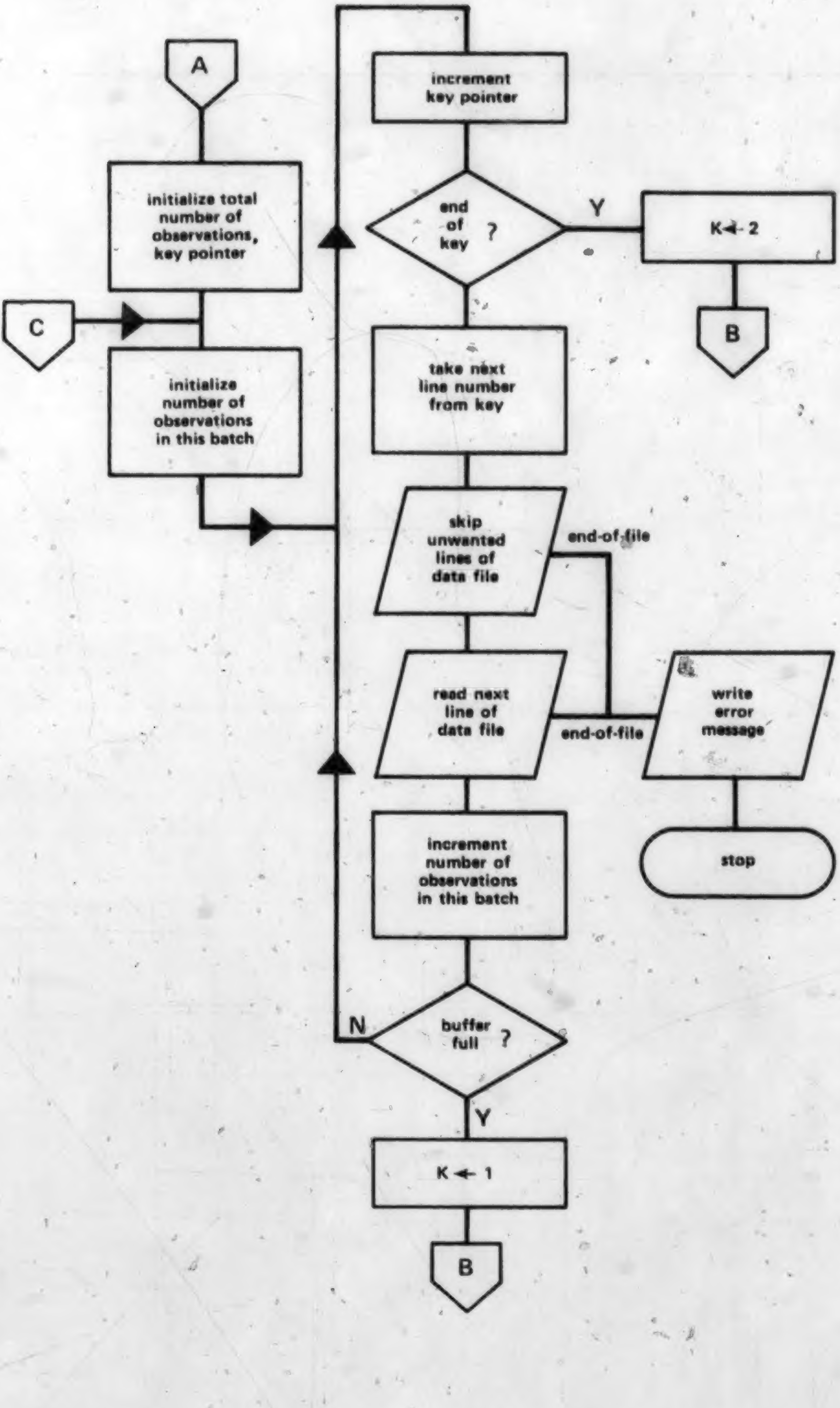
Output is written unformatted on this file as a header record followed by data records. Each header record contains four fullword integers followed by the 120-byte title. The four integers contain (1) the total number of words in this set of data, (2) the number, NV, of variables read from each input record, (3) the total number of input records read in this set of data, (4) the number of output data records following the header record.

Each data record contains two fullword integers followed by the data from up to 100 input records. The data consists of the variables from the first input record read, followed by the variables from the second input record read, and so on. The two leading integers contain (1) the number, N, of input records included in this output record (maximum 100), and (2) the number of words of data in this output record (=N*NV).

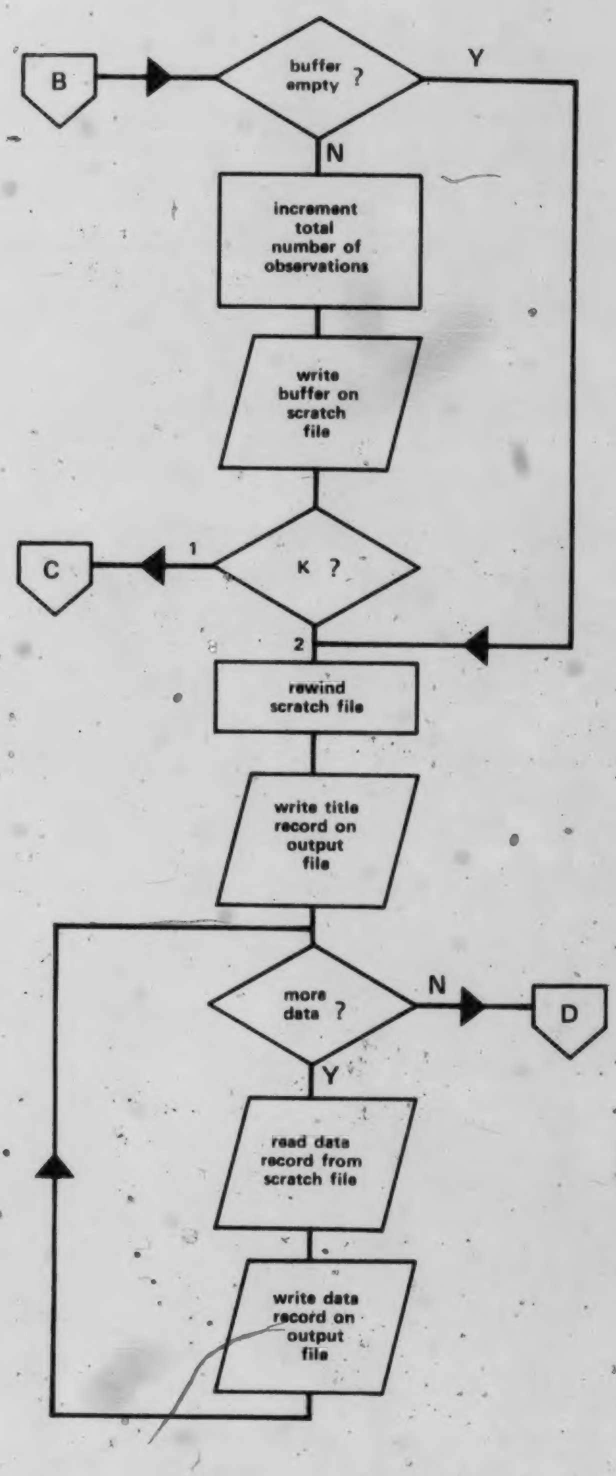
```
C
C
C AREAL-ANALYSIS PACKAGE - PROGRAM "INPUTK1".
C
C
      INTEGER FMT(15),TITLE(30),DATA(1000),CT/'T'/,IC(20)
      REAL RDATA(1000)
      INTEGER*2 KEY(5018)
      LOGICAL*1 TITL/.FALSE./
      EQUIVALENCE (DATA,RDATA)
C
C
    1 FORMAT(15A4)
    2 FORMAT(I10)
    3 FORMAT(A1)
    4 FORMAT(/1X,'**** ERROR IN KEY FILE ****')
    5 FORMAT(/1X,'**** ERROR - END OF DATA FILE ****')
    6 FORMAT(1X,I5,1X,30A4)
    7 FORMAT(I2,15A4,20I2)
   10 FORMAT(1X,I4,' PHYSICAL DATA RECORDS')
   11 FORMAT(A1)
C
C     READ PARAMETERS
C
      READ(5,7)NVAR,FMT,IC
      NVAR=NVAR-1
C
C     IS FIRST LINE A TITLE LINE?
C
      REWIND 1
      READ(1,3)LINTYP
      IF(LINTYP.EQ.CT)TITL=.TRUE.
C
C     READ NEXT KEY
C
  200 CONTINUE
      CALL READK(2,TITLE,NFILE,LENK,KEY,&240,&260)
      WRITE(6,6)LENK,TITLE
  210 REWIND 1
      IF(TITL)READ(1)
      REWIND 7
      LNRO=0
      L=0
      NTOT=0
      NPHR=0
      GOTO300
  240 STOP
  260 WRITE(6,4)
      STOP 12
C
C     READ NEXT BLOCK OF DATA
C
  300 CONTINUE
      N=1
      IB=1
      IE=NVAR
  310 L=L+1
      IF(L.GT.LENK)GOTO350
      LNR=KEY(L)
      NSKIP=LNR-LNRQ-1
      LNRQ=LNR
      IF(NSKIP.EQ.0)GOTO330
      DO 320 I=1,NSKIP
  320 READ(1,11,END=340) DUMMY
  330 READ(1,FMT,END=340) LINTYP, (DATA(I),I=IB,IE)
      N=N+1
      IB=IB+NVAR
      IE=IE+NVAR
      IF(N.LE.100)GOTO310
      K=1
      GOTO360
  340 WRITE(6,5)
      STOP 12
C
C     WRITE NEXT BLOCK OF DATA
C
  350 K=2
  360 N=N-1
      IB=IB-NVAR
      IE=IE-NVAR
      IF(N.EQ.0)GOTO400
      NTOT=NTOT+N
      DO 370 J=2,20,2
      IV=IC(J-1)
      IF(IV.EQ.0)GOTO380
      FACT=1.0/10.0**IC(J)
  365 RDATA(IV)=DATA(IV)*FACT
      IV=IV+NVAR
      IF(IV.LE.IE)GOTO365
  370 CONTINUE
  380 WRITE(7)N,IE,(DATA(I),I=1,IE)
      NPHR=NPHR+1
      GOTO(300,400),K
C
C     RE-WRITE DATA WITH TOTALS
C
  400 REWIND 7
      NN=NVAR*NTOT
      WRITE(8)NN,NVAR,NTOT,NPHR,TITLE
      DO 430 J=1,NPHR
      READ(7)N,IE,(DATA(I),I=1,IE)
  430 WRITE(8)N,IE,(DATA(I),I=1,IE)
      WRITE(6,10)NPHR
  440 GOTO200
      END
```


start
get format (subroutine FORM)
rewind data file
read first line of data file
title ?
Y
set title flag
N
D
read next key
end of file ?
Y
stop
N
rewind data file
title flag set ?
Y
read title
N
rewind scratch file
A

FLOW DIAGRAM FOR PROGRAM INPUTK1 - PART 1


A
initialize total number of observations, key pointer
C
initialize number of observations in this batch
increment key pointer
end of key ?
Y
K ← 2
B
take next line number from key
skip unwanted lines of data file
end-of-file
read next line of data file
end-of-file
write error message
stop
increment number of observations in this batch
N
buffer full ?
Y
K ← 1
B

FLOW DIAGRAM FOR PROGRAM INPUTK1 - PART 2


B
buffer empty ?
Y
N
increment total number of observations
write buffer on scratch file
C
1
K ?
2
rewind scratch file
write title record on output file
more data ?
N
D
Y
read data record from scratch file
write data record on output file

FLOW DIAGRAM FOR PROGRAM INPUTK1 - PART 3

Identification

Program name:

INPUT1

Program title:

Areal-analysis package - data input program.

Author:

John Ramsden

Organization:

Department of Geology, University of Alberta, Edmonton, Alberta, Canada

Date:

1 October 1974

Program version number:

1

Application

Purpose:

To read specified variables from an input file and write the data in a convenient form for use by other programs in the areal-analysis package.

Method:

The program accepts as parameters the number of variables to be read from each record of the input file and the format for reading those variables. The first field in the format specification must refer to a line-type character and must be specified as A1. The given number of variables must include the line-type character. Since the data array is declared as integer, and some FORTRAN input routines will not allow data to be read into an integer variable with an "F" format code (that is, as a real number), all input data must be in integer form and all fields must be specified with an "I" format code. If it is desired to convert

Identification

Program name:

INPUT1

Program title:

Areal-analysis package - data input program.

Author:

John Ramsden

Organization:

Department of Geology, University of Alberta, Edmonton, Alberta, Canada

Date:

1 October 1974

Program version number:

1

Application

Purpose:

To read specified variables from an input file and write the data in a convenient form for use by other programs in the areal-analysis package.

Method:

The program accepts as parameters the number of variables to be read from each record of the input file and the format for reading those variables. The first field in the format specification must refer to a line-type character and must be specified as A1. The given number of variables must include the line-type character. Since the data array is declared as integer, and some FORTRAN input routines will not allow data to be read into an integer variable with an "F" format code (that is, as a real number), all input data must be in integer form and all fields must be specified with an "I" format code. If it is desired to convert integers to real numbers before they are placed in the output file, an additional set of parameters may be added to the parameter card containing the input format. These parameters specify which input variables are to be converted to real numbers and how many decimal places each should have.

At each data record in the input file, the program reads the specified number of one-word variables using the specified format. The data are written unformatted onto the output file as a header record followed by data records. Each output data record except the last always contains the data from 100 input records.

The output data records are first written onto a scratch file. After all the input records have been read, the title record is written onto the output file, and the output data records are then copied from the scratch file to the output file.

After each set of input records (terminated by a delimiter record) has been processed, the program attempts to read another title record. If an end-of-file condition results, the program terminates normally. If another title record is read, the program continues as before to process another set of input records. If the next record is not a title record, the program stops after writing an error message.

Data restrictions:

Maximum length of the given format is 60 characters. Maximum number of variables read from each input record is 10.

Timing:

t=0.0035pv; t=CPU time in seconds; p=total number of data points; v=number of variables.

Implementation

Original installation:

Organization: University of Alberta Computing Services.
Computer: IBM 360 model 67 (1024K).
I/O devices: card, disc, tape, typewriter terminals and display terminals.
Software: Michigan Terminal System (MTS), a virtual-

storage time-sharing system.

Source language:

FORTRAN IV

Subroutines and function subprograms required:

None.

Size:

6336 bytes.

Requirements for I/O devices:

I/O unit no.	read/ write	Maximum FORTRAN record length (bytes)	device control statements	suggested device
1	read	determined by given format	rewind	tape; disc.
5	read	102 formatted	none	cards.
6	write	35 formatted	none	line printer.
7	read, write	8+400*nv, unformatted	rewind	tape; disc.
8	write	8+400*nv, unformatted	none	tape; disc.

nv=number of variables.

Usage

Input:

I/O unit 1: input data file.

The file contains one or several sets of data records. Each set of data records must be preceded by a title record and followed by a delimiter record. The title record must contain the letter "T" in column 1 and the title in columns 2-121. The data records may contain any character except the letter "T" as the line-type character. The delimiter record must contain the letter "T" as the line-type character.

I/O unit 5: parameters.

One recording containing

columns	contents
1-2	number of variables to be read
3-62	format
63-102	up to 20 2-digit numbers. These numbers are interpreted in pairs: the first number of each pair is taken as the number of a variable to be converted to a real number, and the second number of each pair is taken as the number of decimal places. For example, the pair of numbers " 4 3" indicates that the fourth variable (not counting the line-type character) is to be divided by 10**3 and the result placed in the output record as a real number.

Output:

I/O unit 6: messages.

The program writes the number of output data records. Error messages are self-explanatory.

I/O unit 7: scratch file.

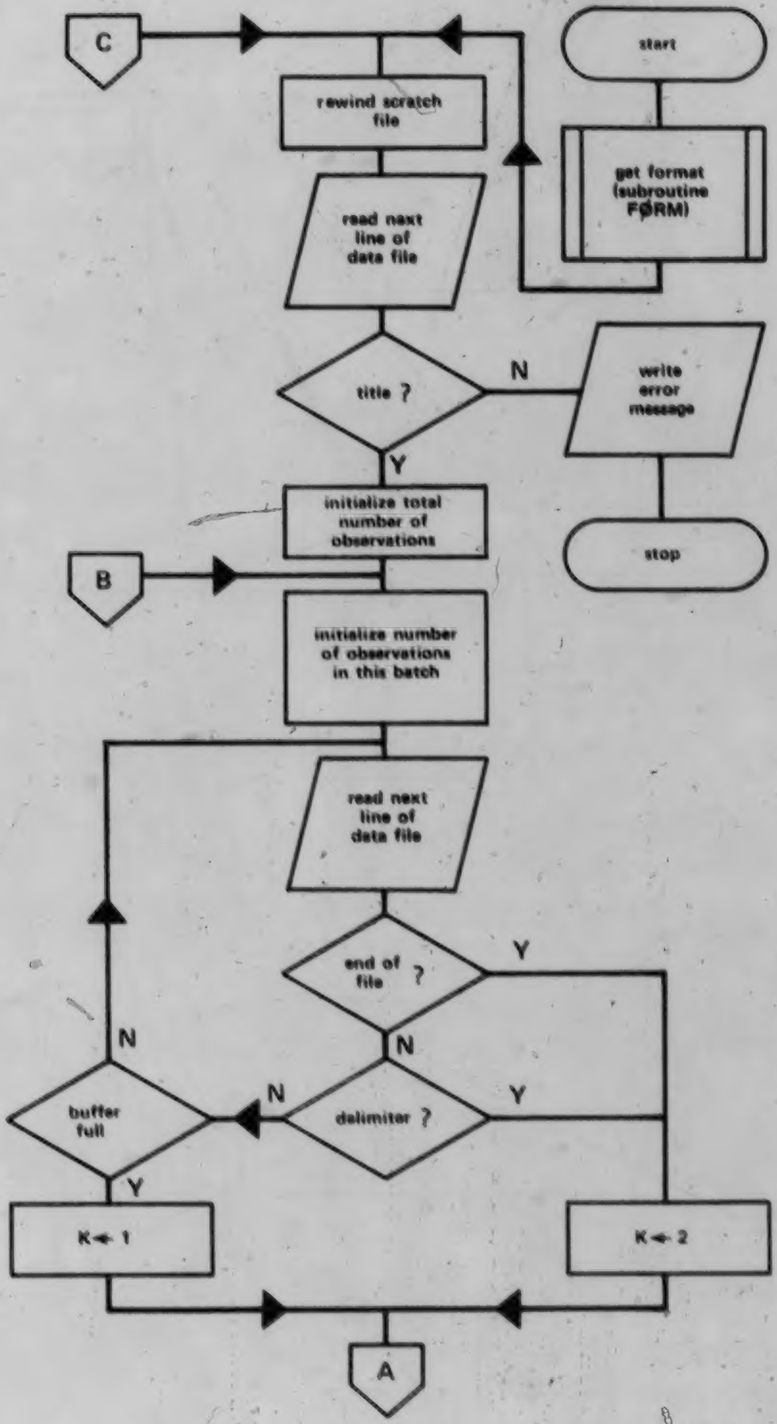
Output data records are stored in this file until they are copied to the output file.

I/O unit 8: output file.

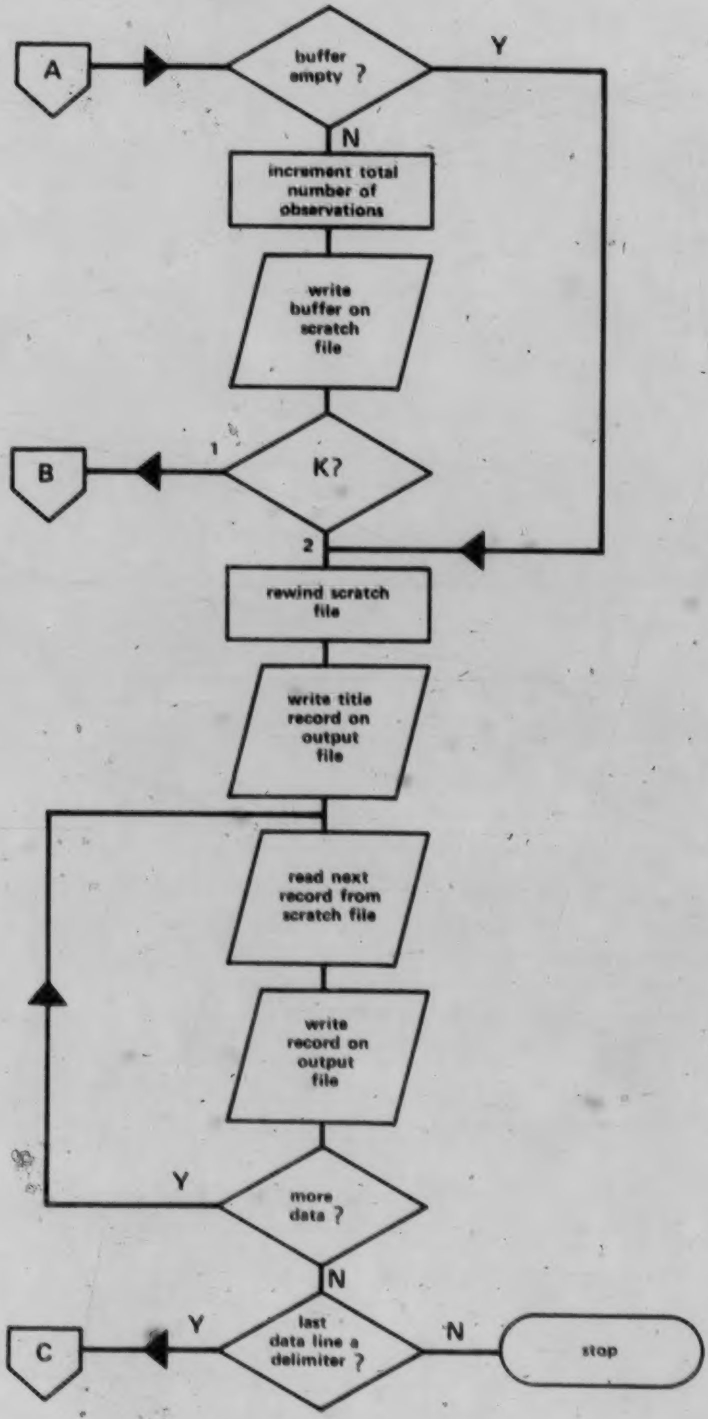
Output is written unformatted on this file as a header record followed by data records. Each header record contains four fullword integers followed by the 120-byte title. The four integers contain (1) the total number of words in this set of data, (2) the number, NV, of variables read from each input record, (3) the total number of input records read in this set of data, (4) the number of output data records following the header record.

Each data record contains two fullword integers followed by the data from up to 100 input records. The data consists of the variables from the first input record read, followed by the variables from the second input record read, and so on. The two leading integers contain (1) the number, N, of input records included in this output record (maximum 100), and (2) the number of words of data in this output record (=N*NV).

```
C
C
C AREA-ANALYSIS PACKAGE - PROGRAM "INPUT1".
C
C
      INTEGER FMT(15),CT/'T'/,TITLE(30)
     *,DATA(1000),IC(20)
      REAL RDATA(1000)
      EQUIVALENCE (DATA,RDATA)
C
C
    1 FORMAT(15A4)
    2 FORMAT(I10)
    3 FORMAT(A1,30A4)
    4 FORMAT(1X,30A4)
    7 FORMAT(1X,'***** ERROR - TITLE MISSING IN DATA FILE *****')
    8 FORMAT(I2,15A4,20I2)
    9 FORMAT(1X,30A4)
   10 FORMAT(1X,I4,' DATA POINTS',5X,I4,' PHYSICAL RECORDS')
   11 FORMAT(1X,I4,' VARIABLES'/1X,'FORMAT=',25A4)
C
C
      READ(5,8)NVAR,FMT,IC
      WRITE(6,11)NVAR,FMT
      NVAR=NVAR-1
C
C     IS NEXT LINE A TITLE?
C
  140 REWIND 7
      READ(1,3,END=500)LINTYP,TITLE
      WRITE(6,9)TITLE
      IF(LINTYP.NE.CT)GOTO900
  200 NTOT=0
      NPHR=0
C
C     READ AND WRITE NEXT BATCH OF DATA
C
  220 N=1
      IB=1
      IE=NVAR
  240 READ(1,FMT,END=290)LINTYP,(DATA(I),I=IB,IE)
      IF(LINTYP.EQ.CT)GOTO290
      N=N+1
      IB=IB+NVAR
      IE=IE+NVAR
      IF(N.LE.100)GOTO240
      K=1
      GOTO300
C
  290 K=2
  300 N=N-1
      IB=IB-NVAR
      IE=IE-NVAR
      IF(N.EQ.0)GOTO360
      NTOT=NTOT+N
      DO 340 J=2,20,2
      IV=IC(J-1)
      IF(IV.EQ.0)GOTO350
      FACT=1.0/10.0**IC(J)
  330 RDATA(IV)=DATA(IV)*FACT                                          INPT 600
      IV=IV+NVAR                                                       INPT 610
      IF(IV.LE.IE)GOTO330                                              INPT 620
  340 CONTINUE                                                         INPT 630
  350 WRITE(7)N,IE,(DATA(I),I=1,IE)                                    INPT 640
      NPHR=NPHR+1                                                      INPT 650
      GOTO(220,360),K                                                  INPT 660
C                                                                      INPT 670
C     RE-WRITE DATA WITH TOTALS                                        INPT 680
C                                                                      INPT 690
  360 REWIND 7                                                         INPT 700
      NN=NVAR*NTOT                                                     INPT 710
      WRITE(8)NN,NVAR,NTOT,NPHR,TITLE                                  INPT 720
      DO 390 J=1,NPHR                                                  INPT 730
      READ(7)N,IE,(DATA(I),I=1,IE)                                     INPT 740
  390 WRITE(8)N,IE,(DATA(I),I=1,IE)                                    INPT 750
      WRITE(6,10)NTOT,NPHR                                             INPT 760
  400 IF(LINTYP.EQ.CT)GOTO140                                          INPT 770
  500 STOP                                                             INPT 780
C                                                                      INPT 790
  900 WRITE(6,7)                                                       INPT 800
      STOP12                                                           INPT 810
C                                                                      INPT 820
      END                                                              INPT 830
```


C
start
rewind scratch file
get format (subroutine FØRM)
read next line of data file
title ?
N
write error message
stop
Y
initialize total number of observations
B
initialize number of observations in this batch
read next line of data file
end of file ?
Y
N
delimiter ?
Y
N
buffer full
N
Y
K ← 1
K ← 2
A

FLOW DIAGRAM FOR PROGRAM INPUT1 - PART 1


A
buffer empty ?
Y
N
increment total number of observations
write buffer on scratch file
K?
1
B
2
rewind scratch file
write title record on output file
read next record from scratch file
write record on output file
more data ?
Y
N
last data line a delimiter ?
Y
C
N
stop

FLOW DIAGRAM FOR PROGRAM INPUT1 - PART 2

SECTION 1 : PROGRAM IDENTIFICATION

PROGRAM TITLE:

Search/retrieval program for identical-record data files

PROGRAM CODE NAME:

KEY4

WRITER:

John Ramsden

ORGANIZATION:

Department of Geology, University of Alberta, Edmonton, Alberta

DATE:

7 December 1973

VERSION:

Zero

SOURCE LANGUAGE:

FORTRAN IV

AVAILABILITY:

Available from Dr. G. Herget
Elliot Lake Laboratory
CANMET, EMR
P.O. Box 100
Elliot Lake, Ontario
P5A 2J6

ABSTRACT:

This program searches a data file containing identical-format records for those records satisfying a given combination of conditions. It either copies those records into a separate file, or simply outputs the numbers of those records.

SECTION 2: ENGINEERING DOCUMENTATION

NARRATIVE DESCRIPTION:

This program will search a data file for those records satisfying a given combination of conditions. It will either store the numbers of those records, or move those records into a separate file.

METHOD OF SOLUTION:

The program performs a series of discrete operations, each operation being initiated by a parameter card. These operations are of two kinds: those that involve a condition ("SCAN" operations), and those that do not. If the operation is of the first kind, the program examines the data file and stores the numbers of those data records that satisfy the given condition (involving one field of the data record). Depending on the parameters, the program will examine either the entire data file or a subset of the data file identified in a previous operation.

Operations that do not involve a condition require some manipulation of the results of previous operations to produce a new set of record numbers. The data file is not examined.

On completion of any operation it is possible to (1) save permanently the set of record numbers produced by the operation, and/or (2) actually retrieve the data records whose numbers belong to that set and place them in a separate file. It is also possible to input sets of record numbers saved from previous runs.

PROGRAM CAPABILITIES:

The records of the data file must all have the same length and format.

The program is presently designed for a data file containing 112-byte records. The present limit on the size of a subset of the data file that can be defined is 5000 records. These limits may be changed as described below.

If the program is to be used on a file whose records are not 112 bytes long, the sizes of the vectors "line" and"bline" in subroutine "nretr" must be changed, as well as format statement number 1 in the same subroutine.

To change the maximum size of a subset of a data file that can be handled, the dimensions of the variables "key1", "key2" and "key3" in all program segments except "posn", "rkey", "read" and "write" must be changed to the new maximum plus 18. The dimension of "key" in subroutines "read" and "write" must be similarly changed.

If the sizes of these arrays are changed, the maximum length of unformatted records temporarily stored on the device assigned to I/O unit 2 will also change, and will be equal to the length of the array "key3" plus one full word (plus words used by the system).

DATA INPUTS:

There are three essential inputs. One is the data file. The second is a table identifying the named fields within the data record. The third is the set of instructions giving the conditions to be searched for.

I/O unit 1: The data file

The records of the data file must all have the same length and format. The program may be modified for any record length as described above. Within the data record, integer fields of any length, and alpha fields of up to four characters are defined by field-definition cards.

I/O unit 3: Field-definition cards

These have the following format.

col	contents
1-8	field name

9-34	blank
35-43	FORTRAN format defining field e.g. (51X,I5) ('I' or 'A' format only)

There may be from 1 to 99 field-definition cards, preceded by one card containing the number of field-definition cards right-justified in columns 1 and 2.

I/O unit 4: Input sets of record numbers

As generated by the program.

I/O unit 5: Parameter cards

These contain fixed parameter fields as follows.

number	name	columns
1	label 1	1-8
2	operation	10-13
3	label2	15-22
4	field name	24-31
5	relational operator	33-34
6	alpha value	36-39
7	first numeric value	41-48
8	second numeric value	50-57
9	label3	59-66
10	'save' character	68
11	'retrieve' character	69

All fields are left-justified alphanumeric except 7 and 8 which are right-justified integer.

Order of Input:

If I/O units 3, 4, and 5 are all assigned to the same input device, the order of input must be as follows: field-definition cards first, followed by parameter cards. One and only one previously saved set of record numbers must immediately follow each parameter card that specifies a "READ" operation.

PROGRAM OPTIONS:

There are 6 operation codes: SCAN, READ, COMM, PLUS, LESS,STOP.

SCAN

The 'scan' operation involves examining the data file or some subset of it and storing the numbers of those data records satisfying a given condition. 'label2' identifies the subset of the data to be examined. If 'label2' is blank the entire data file is examined. 'field name' is one of the field names defined on the field-definition cards. 'relational operator' is one of the following:

EQ	(equal to)	NE	(not equal to)
LT	(less than)	NL	(not less than)
GT	(greater than)	NG	(not greater than)
GE	(greater than or equal to)	LE	(less than or equal to)
IR	(in range)	OR	(outside range)

If 'field name' was defined with an 'A' format code, a character string is to be placed in parameter field 6, and relational operators other than 'EQ' and 'NE' should not be used. If 'field name' was defined with an 'I' format code, then a single integer is to be placed in parameter field 7 for all relational operators other than 'IR' and

'OR', while for these relational operators the lower and upper limits of the range (respectively) are to be placed in parameter fields 7 and 8. The range is considered to include the lower limit and exclude the upper limit.

READ

This operation code causes the program to read from I/O unit 4 a set of record numbers (saved from a previous run).

COMM

This code causes the generation of a set of record numbers consisting of those record numbers that belong to both of the sets identified by 'label 1' and label2'.

PLUS

This code causes the generation of a set of record numbers consisting of those record numbers that belong to either of the two sets identified by 'label 1' and 'label2'.

LESS

This code causes the generation of a set of record numbers consisting of those record numbers that belong to the set identified by 'label 1' but do not belong to the set identified by 'label2'.

STOP

This code causes the program to stop.

For all operation codes except STOP, the following parameters have the same function. 'label3' is the label to be associated with the set of record numbers generated by the current operation. If the 'save character' is non-blank the set of record numbers generated by the current operation will be saved. If the 'retrieve character' is non-blank the data records corresponding to the set of record numbers generated by the current operation will be copied into a separate file.

PRINTED OUTPUT:

The program prints a message as each operation is completed.

OTHER OUTPUTS:

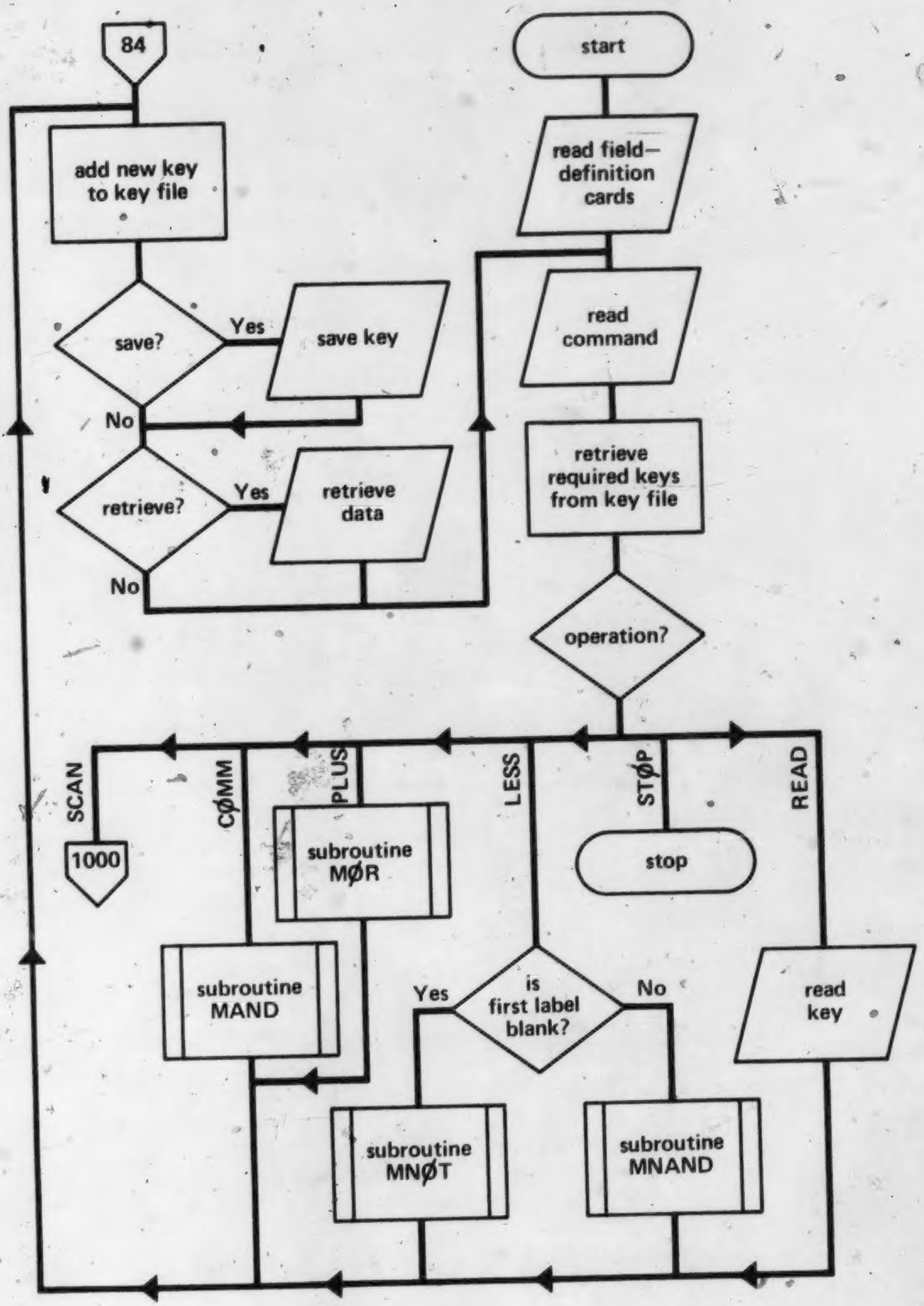
I/O unit 7: saved sets of record numbers

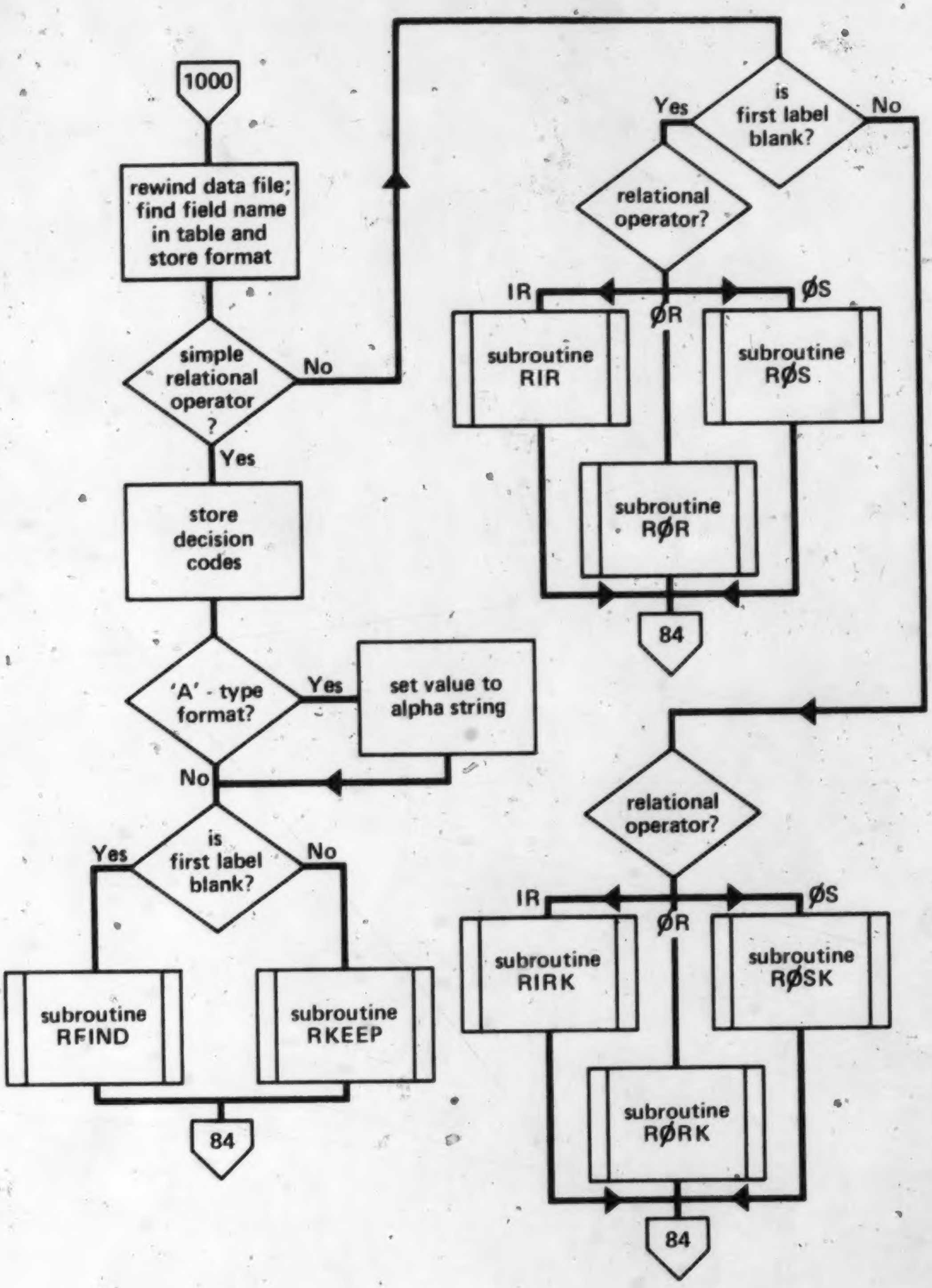
Each set of record numbers saved is written out in 80-byte lines, numbered sequentially in columns 78-80. The first two lines contain the label, the number of records in the data file, and the number of record numbers in the set. Each subsequent line contains record numbers in 18 adjacent 4-column fields in columns 1 to 72. Unused fields on the last line contain zeros.

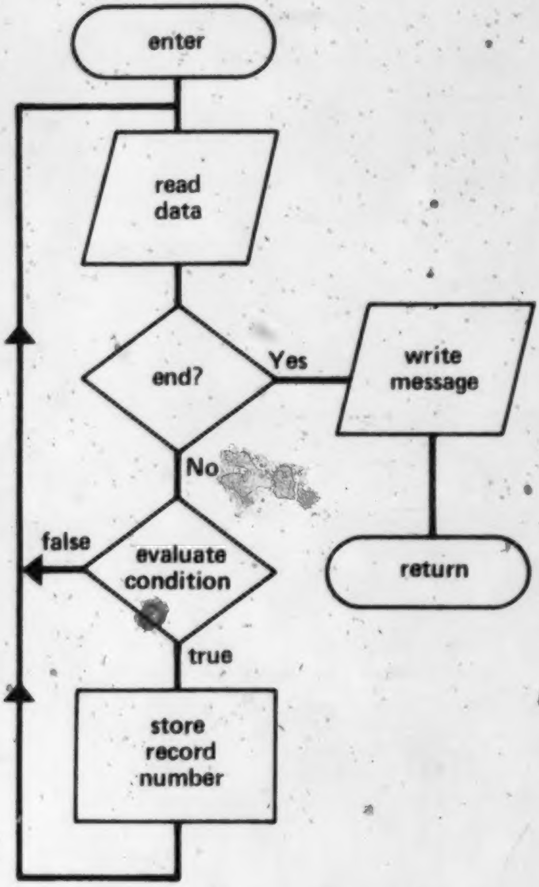
I/O unit 8: retrieved data records

For each subset of the data file retrieved, the program writes on I/O unit 8 one record containing the associated label, followed by the retrieved data records, followed by one blank record.

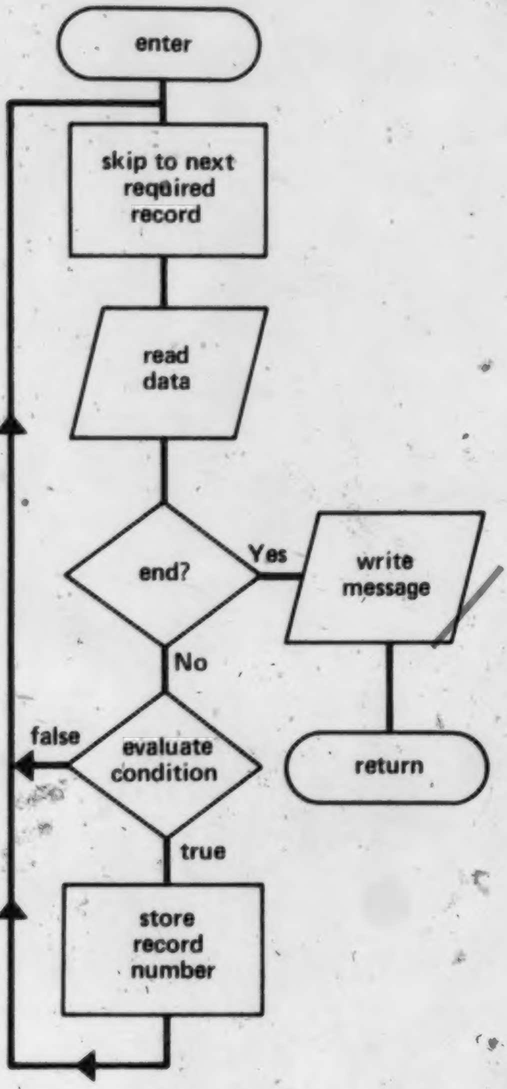
FLOW CHART:


84
start
add new key to key file
read field-definition cards
save?
Yes
save key
No
read command
retrieve?
Yes
retrieve data
No
retrieve required keys from key file
operation?
SCAN
CØMM
PLUS
LESS
STØP
READ
1000
subroutine MØR
stop
subroutine MAND
Yes
is first label blank?
No
read key
subroutine MNØT
subroutine MNAND

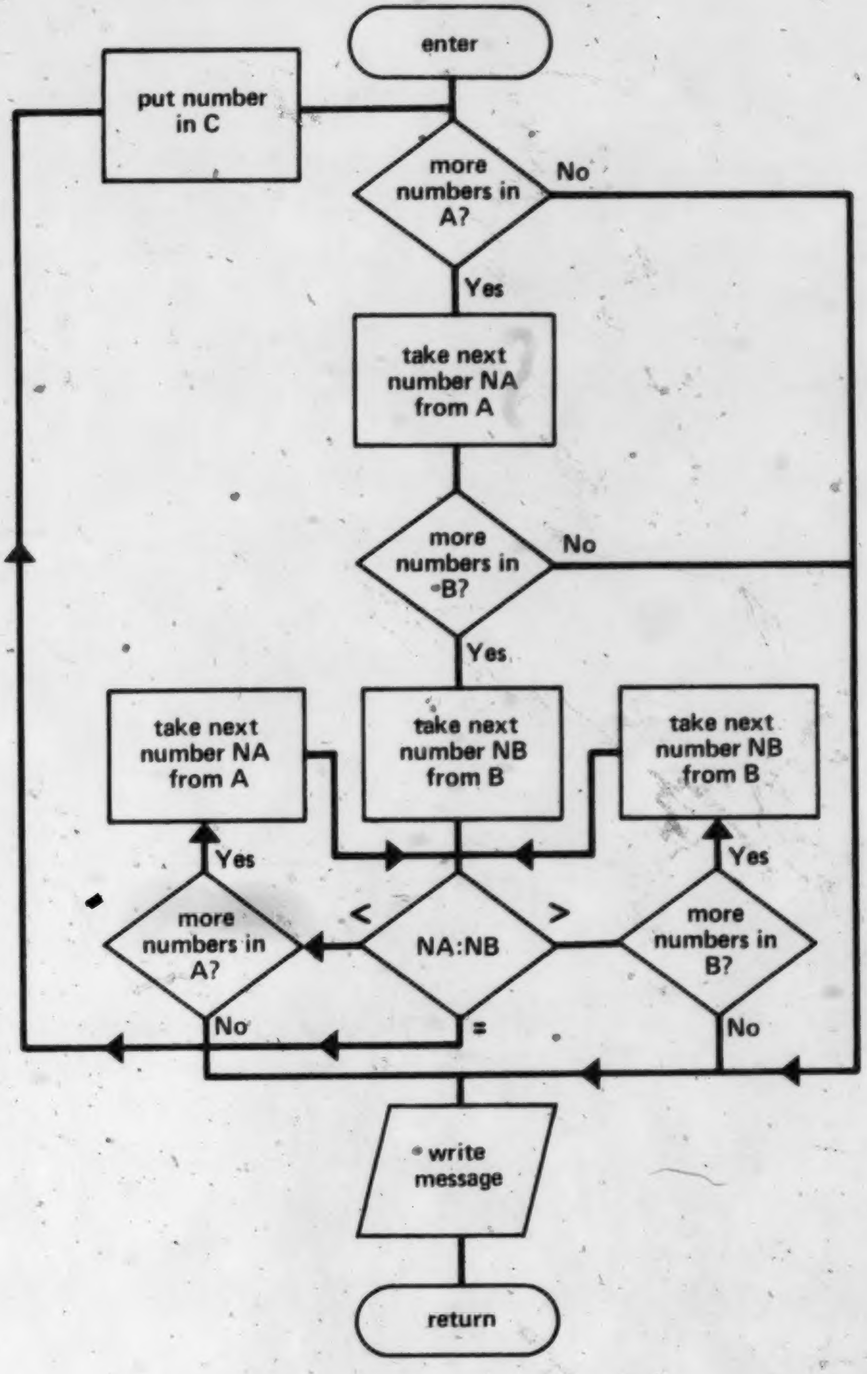

1000
rewind data file; find field name in table and store format
simple relational operator ?
No
Yes
store decision codes
'A' - type format?
Yes
set value to alpha string
No
is first label blank?
Yes
No
subroutine RFIND
subroutine RKEEP
84
is first label blank?
Yes
No
relational operator?
IR
ØR
ØS
subroutine RIR
subroutine RØS
subroutine RØR
84
relational operator?
IR
ØR
ØS
subroutine RIRK
subroutine RØSK
subroutine RØRK
84


enter
read data
end?
Yes
write message
No
return
false
evaluate condition
true
store record number

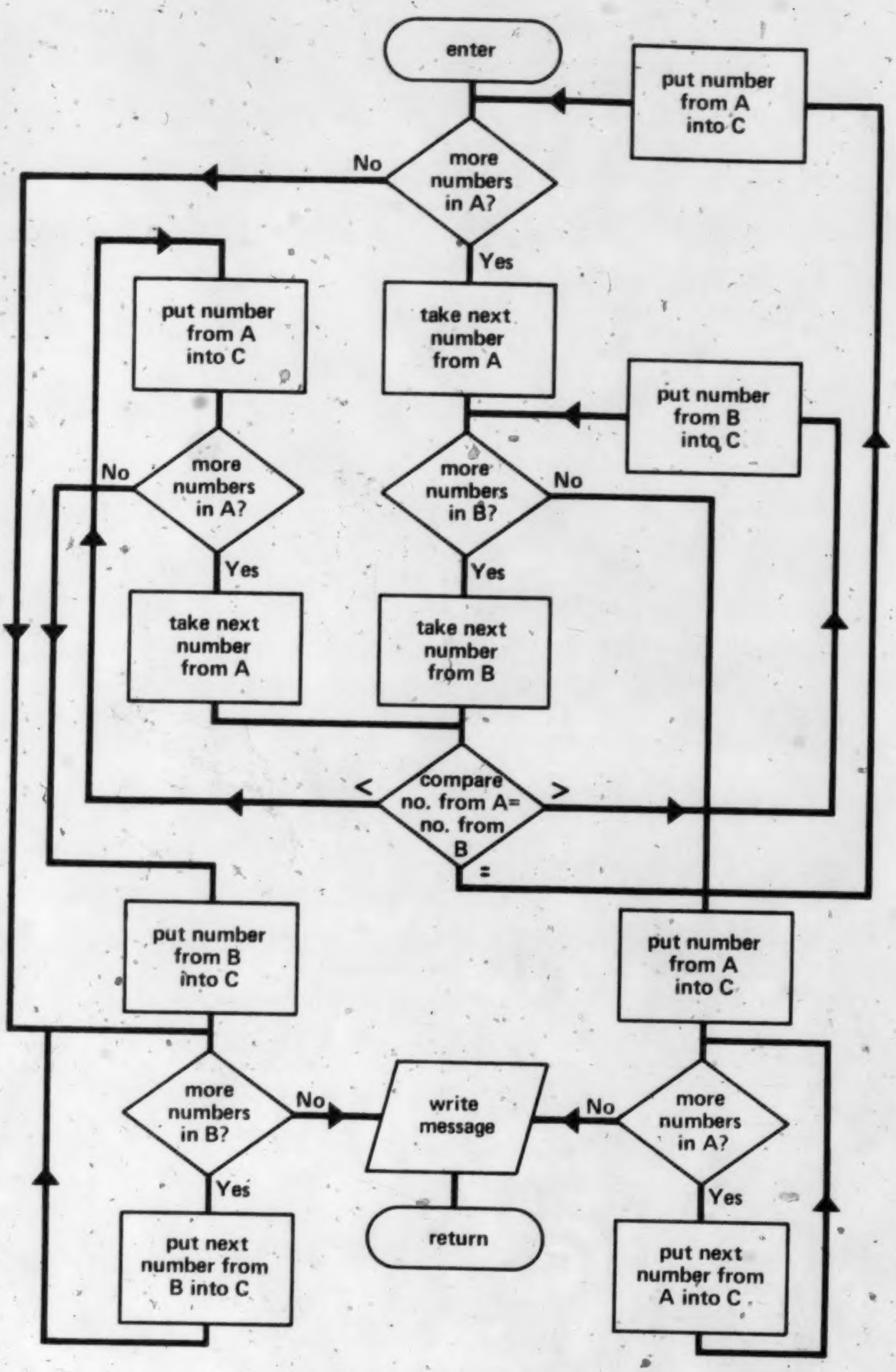
subroutines
RFIND, RIR, RØR, RØS

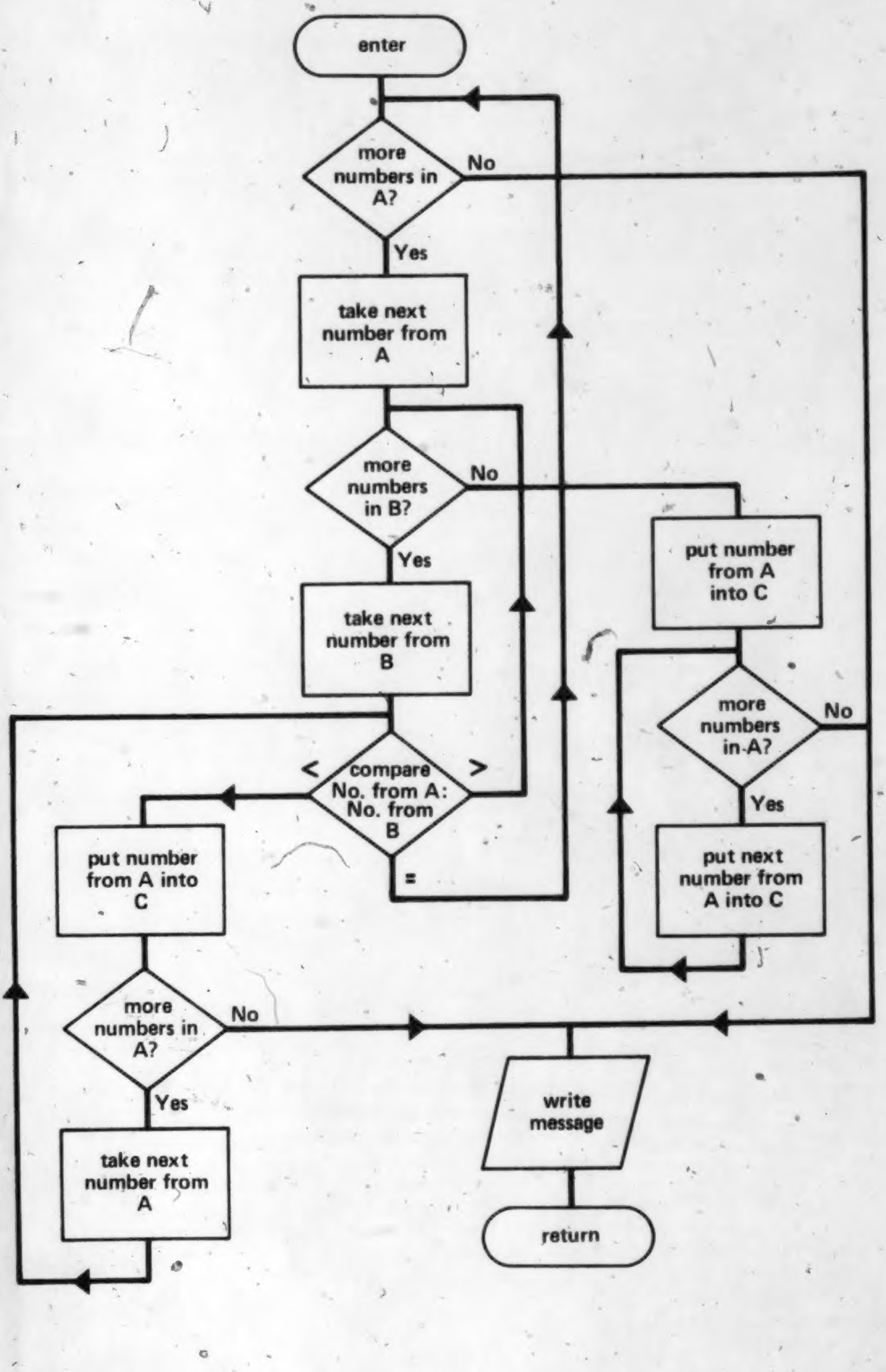

enter
skip to next required record
read data
end?
Yes
write message
No
return
false
evaluate condition
true
store record number

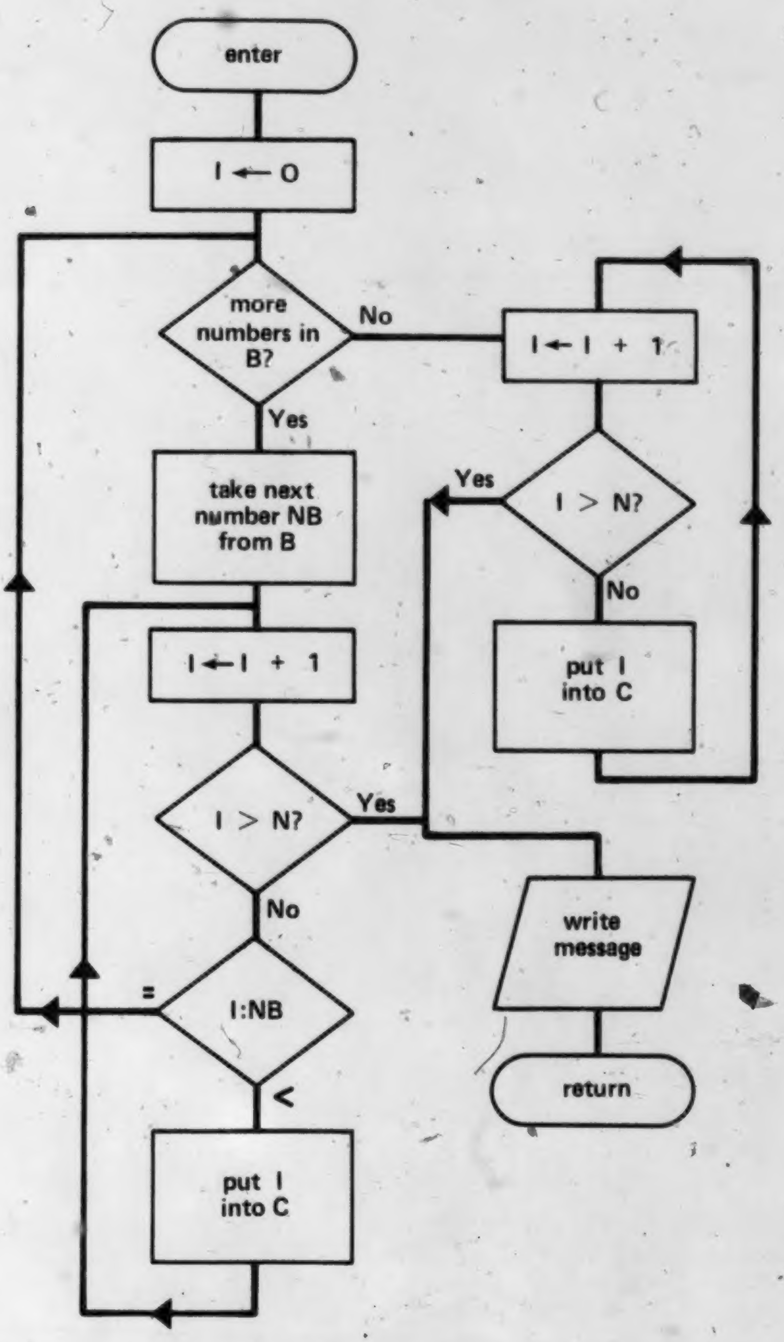
subroutines
RKEEP, RIRK, RØRK, RØSK


enter
put number in C
more numbers in A?
No
Yes
take next number NA from A
more numbers in B?
No
Yes
take next number NA from A
take next number NB from B
take next number NB from B
Yes
Yes
more numbers in A?
<
NA:NB
>
more numbers in B?
No
=
No
write message
return

subroutine MAND


enter
put number from A into C
No
more numbers in A?
Yes
take next number from A
put number from A into C
put number from B into C
No
more numbers in A?
more numbers in B?
No
Yes
Yes
take next number from A
take next number from B
<
compare no. from A= no. from B
>
=
put number from B into C
put number from A into C
more numbers in B?
No
write message
No
more numbers in A?
Yes
Yes
return
put next number from B into C
put next number from A into C


enter
more numbers in A?
No
Yes
take next number from A
more numbers in B?
No
Yes
take next number from B
put number from A into C
compare No. from A: No. from B
<
>
=
put number from A into C
more numbers in A?
No
Yes
more numbers in A?
No
Yes
put next number from A into C
take next number from A
write message
return


enter
I ← 0
more numbers in B?
No
Yes
take next number NB from B
I ← I + 1
I > N?
Yes
No
I:NB
=
<
put I into C
I ← I + 1
Yes
I > N?
No
put I into C
write message
return

subroutine MNØT

SAMPLE RUN:

Table of data names and formats:

```
37
YEAR                              (001X,I1)
GEOL                              (002X,A2)
TRAVERSE                          (004X,I4)
DOMAIN                            (008X,A2)
FORMATIO                          (010X,A3)
LEVEL                             (013X,I4)
LOCATION                          (017X,A3)
TREND                             (020X,I3)
PLUNGE                            (023X,I3)
LENGTH                            (026X,I4)
NOBS                              (030X,I2)
SCALE                             (032X,I1)
SEQNO                             (033X,I2)
DISTANCE                          (035X,I4)
EASTING                           (039X,I6)
NORTHING                          (045X,I6)
ELEVATIO                          (051X,I5)
TYPE                              (056X,A2)
DIPDIR                            (058X,I3)
DIP                               (061X,I3)
COSA                              (064X,I5)
COSB                              (069X,I5)
COSG                              (074X,I5)
WEIGHT                            (079X,I3)
SIZE                              (082X,A1)
SPACING                           (083X,A1)
FILLING1                          (084X,A1)
FILLING2                          (085X,A1)
WATER                             (086X,A1)
WAVES                             (087X,I3)
LINETYPE                          (090X,A2)
LINA                              (092X,I5)
LINB                              (097X,I5)
LING                              (102X,I5)
LITHOLOG                          (107X,A3)
HARDNESS                          (110X,A2)
MINE.LEV                          (013X,I2)
↑                                 ↑
1                                 35
```

Search and retrieval instructions:

```
SCAN          LINETYPE NE                         LINEARS  XX
SCAN LINEARS  DIP      IR             80       90 SET1     XX
STOP
↑    ↑        ↑        ↑              ↑        ↑  ↑        ↑
10   15       24       33             47       56 59       68
```

Input Data File:

```
TXFILEVJN
D2DM0002010R45920   222 -21001680 1  2478381496430259422JN 82 60-1205-8576 5000 15J   3171FD-9067 3001 2962SDSR5
D2DM0002010R45920   222 -21001680 2  2878391196429959422JN352 87-9889 1390  523 16J   3164FD 1480 8944 4220SDSR5
D2DM0002010R45920   222 -21001680 4  7478379196426559424JN 95 80  858-9811 1736 17J   3163FD -151 1730 9848SDSR5
D2DM0002010R45920   222 -21001680 9 1237837489642295942 5JN105 85 2578-9623  872 22J   3       0    0    0SDSR5
D2DM0002010R45920   222 -2100168010 1557837269642055942 6JN102 85 2071-9744  872 20J   3172FD 3175 1512 9361SDSR5
D2DM0002010R45920   222 -2100168011 1807837109641865942 7JN127 80 5927-7865 1736115J F 3174FD-7862-5181 3368SDSR5
D2DM0002010R45920   222 -2100168023 3677835719640335943 5JK102 85 2071-9744  872 20K   3       0    0    0SDSR5
D2DM0002010R45920   222 -2100168025 4167835529640115943 6JN 91 69  163-9334 3584 17K   3170FD 9955  486  814SDSR5
D2DM0002010R45920   222 -2100168026 4177835519640105943 6JN104 82 2396-9609 1392 22J F 3166FD-9695-2292  863SDSR5
D2DM0002010R45920   222 -2100168027 4347835409639985943 6JN102 74 1999-9403 2756 21J F 3162FD  453 2899 9560SDSR5
D2DM0002010R45920   222 -2100168031 6137834209638665944 2JN156 86 9113-4057  698 25J   3       0    0    0SDSR5
D2DM0002010R45920   222 -2100168032 5977834319638775944 2JN 73 54-1544-7942 5878 16J ME4173FD 9871-1501  564SDSR5
D2DM0002010R45920   222 -2100168033 5857834399638865944 1JN107 76 2837-9279 2419 25K   3156FD 9335 3250 1518SDSR5
D2DM0002010R45920   222 -2100168034 6147834199638645944 2JN107 60 2532-8282 5000 29K   3164FD 9506 3099  302SDSR5
D2DM0002010R45920   222 -2100168036 6607833899638305944 4JN 94 80  687-9824 1736 17K   4167FD-9976 -698    0SDSR5
D2DM0002010R45920   222 -2100168037 6697833839638235944 4JN164 85 9576-2746  872 19K   3160FD-2960-9425 1730SDSR5
D2DM0002010R45920   222 -2100168038 7147833539637905944 6JN101 87 1905-9803  523 20K   3171FD-9788-1856  870SDSR5
D2DM0002010R45920   222 -2100168039 7477833309637655944 7JN142 75 7612-5947 2588 57K   3177FD-6412-6298 4385SDSR5
D2DM0002010R45920   222 -2100168048 8567832589636845945 1JN 98 68 -324-9266 3746 16J   3173FD-9994  349    0SDSR5
D2DM0002010R45920   222 -2100168049 8677832509636765945 1JN150 80 8529-4924 1736 32K   3159FD-3257-2439 9131SDSR5
D2DM0002010R45920   222 -2100168052 9027832279636505945 2JN 92 80  344-9842 1736 16J   3       0    0    0SDSR5
D2DM0002010R45920   222 -2100168053 9297832099636305945 3JN322 70-7405 5785 3420 66J   2164FD 6644 7067 2432SDSR5
D2DM0002010R45920   222 -2100168054 9307832089636295945 3JN157 85 9170-3892  872 24J   3175FD-3982-9803 2578SDSR5
D2DM0002010R45920   222 -2100168055 9527831939636135945 4JN102 80 2048-9533 1736 21J   3       0    0    0SDSR5
D2DM0002010R45920   222 -2100168056 9627831379636065945 5JN 82 75-1344-9565 2588 14J   3       0    0    0SDSR5
D2DM0002010R45920   222 -2100168057 9687831839636015945 5JN 77 75-2173-9412 2588 13I   3       0    0    0SDSR5
D2DM0002010R45920   222 -2100168058 9807831759635925945 5JN 92 85  348-9956  872 16K   3       0    0    0SDSR5
D2DM0002010R45920   222 -2100168060 9827831739635915945 5JN207 85 8876 4523  872 10J   3170FD 4000-8508 3407SDSR5
D2DM0002010R45920   222 -2100168061 9837831739635905945 5JN 92 90  349-9994    0 16J   3163FD-9994 -349    0SDSR5
D2DM0002010R45920   222 -2100168062 9957831659635815945 6JN 87 90 -523-9986    0 14J   3       0    0    0SDSR5
D2DM0002010R45920   222 -2100168063100078316196357759456JN197 80 9418 2879 1736 11I   2       0    0    0SDSR5
D2DM0003010R45920   227  0 21124017 2117829869634865953 8JN162 82 9418-3060 1392 24J   3       0    0    0SDSR5
D2DM0003010R45920   227  0 21124018   07831409636305953 8JN 91 83  173-9924 1219 14 I   174FD 9652  484 2569SDSR5
D2DM0003010R45920   227  0 21124019   07831409636305953 8JN 42 90-7431-6691    0 10 I   155FD-6399 7107 2924SDSR5
D2DM0003010R45920   227  0 21124020   07831409636305953 8JN 89 86 -174-9974  698 13 I             0    0    0SDSR5
D2DM0003010R45920   227  0 21124021   07831409636305953 8JN201 87 9323 3579  523 11 I             0    0    0SDSR5
D2DM0003010R45920   227  0 21124023   07831409636305953 8JN 93 93  519-9912 1219 15 I             0    0    0SDSR5
D2DM0003010R45920   227  0 21124024   07831409636305953 8JN 25 90-9063-4226    0 11 I             0    0    0SDSR5
```

Printed output:

```
          SCAN          LINETYPE NE              0        0 LINEARS  XX

 END OF DATA

    38 LINES READ        23 LINE NUMBERS IN KEY
 NEW KEY ADDED TO FILE
 LINEARS  HAS BEEN SAVED
 LINEARS  HAS BEEN RETRIEVED


          SCAN LINEARS  DIP      IR              80       90 SET1     XX

 END OF DATA

 OLD KEY CONTAINED   23 LINE NUMBERS     NEW KEY CONTAINS   12 LINE NUMBERS
 NEW KEY ADDED TO FILE
 SET1      HAS BEEN SAVED
 SET1      HAS BEEN RETRIEVED


          STOP                                   0        0
```

Output "keys":

```
LINEARS
                                                                               1
                                                      38      23               2
   1   2   3   5   6   8   9  10  12  13  14  15  16  17  18  19  20  22       3
  23  28  29  33  34   0   0   0   0   0   0   0   0   0   0   0   0   0       4
SET1
                                                                               1
                                                      38      12               2
   2   3   5   6   9  15  16  17  20  23  28  33   0   0   0   0   0   0       3
```

Output retrieved data file:

```
TLINEAPS
D2DM0002010F45920   222 -21001680 1  2478381496430259422JN 82 60-1205-8576 5000 15J    3171FD-9067 3001 2962SDSR5
D2DM0002010F45920   222 -21001680 2  2878361196429959422JN352 87-9889 1390  523 16J    3164FD 1480 8944 4220SDSR5
D2DM0002010F45920   222 -21001680 4  7478378196426559424JN 95 80  658-9611 1736 17J    3163FD -151 1730 9848SDSR5
D2DM0002010F45920   222 -2100168010 15578372696420559426JN102 85 2071-9744  872 20J    3172FD 3175 1512 9361SDSR5
D2DM0002010F45920   222 -2100168011 18078371096418659427JN127 90 5927-7665 1736115J F 3174FD-7862-5181 3368SDSR5
D2DM0002010F45920   222 -2100168025 41678355296401159436JN 91 69  163-9334 3584 17K    3170FD 9955  486  914SDSR5
D2DM0002010F45920   222 -2100168026 41778355196401059436JN104 82 2396-9609 1392 22J F 3166FD-9695-2292  863SDSR5
D2DM0002010F45920   222 -2100168027 43478354096399859436JN102 74 1999-9403 2756 21J F 3162FD  453 2899 9560SDSR5
D2DM0002010F45920   222 -2100168032 59778343196387759442JN 79 54-1544-7942 5878 16J M54178FD 9971-1501  564SDSR5
D2DM0002010F45920   222 -2100168033 56578343996386659441JN107 76 2837-9279 2419 25K    3166FD 9335 3250 1518SDSR5
D2DM0002010F45920   222 -2100168034 61478341996386459442JN107 60 2532-8282 5000 29K    3164FD 9506 3089 1302SDSR5
D2DM0002010F45920   222 -2100168036 66078338996353059444JN 94 80  687-9824 1736 17K    4167FD-9976 -699    0SDSR5
D2DM0002010F45920   222 -2100168037 66978338396382359444JN164 85 9576-2746  872 19K    3160FD-2860-9425 1730SDSR5
D2DM0002010F45920   222 -2100168038 71478335396379059446JN101 87 1905-9803  523 20K    3171FD-9738-1656  870SDSR5
D2DM0002010F45920   222 -2100168039 74778333096376559447JN142 75 7612-5947 2588 57K    3177FD-6412-6298 4365SDSR5
D2DM0002010F45920   222 -2100168048 63678325896368459451JN 98 66 -324-9266 3746 16J    3173FD-9994  349    0SDSR5
D2DM0002010F45920   222 -2100168049 86778325096367659451JN150 90 8529-4924 1736 32K    3159FD-3267-2439 9131SDSR5
D2DM0002010F45920   222 -2100168053 92978320996363059453JN322 70-7405 5785 3420 66J    2164FD 6644 7067 2432SDSR5
D2DM0002010F45920   222 -2100168054 93078320896362959453JN157 85 9170-3892  872 24J    3175FD-3982-8803 2576SDSR5
D2DM0002010F45920   222 -2100168060 98278317396359159455JN207 65 8876 4523  872 10J    3170FD 4000-8508 3407SDSR5
D2DM0002010F45920   222 -2100168061 98378317396359059453JN 92 90  349-9994    0 16J    3163FD-9994 -349    0SDSR5
D2DM0002010F45920   227  0 21124018    078314096363059538JN 91 83  173-9924 1219 14 I    174FD 9652  484 2569SDSR5
D2DM0003010F45920   227  0 21124019    078314096363059538JN 42 90-7431-6691    0 10 I    155FD-6399 7107 2924SDSR5

TSET1
D2DM0002010F45920   222 -21001680 2  2878381196429959422JN352 87-9889 1390  523 16J    3164FD 1480 8944 4220SDSR5
D2DM0002010F45920   222 -21001680 4  7478378196426559424JN 95 80  858-9811 1736 17J    3163FD -151 1730 9848SDSR5
D2DM0002010F45920   222 -2100168010 15578372696420559426JN102 85 2071-9744  872 20J    3172FD 3175 1512 9361SDSR5
D2DM0002010F45920   222 -2100168011 18078371096418659427JN127 80 5927-7865 1736115J F 3174FD-7862-5181 3368SDSR5
D2DM0002010F45920   222 -2100168026 41778355196401059436JN104 82 2396-9609 1392 22J F 3166FD-9695-2292  863SDSR5
D2DM0002010F45920   222 -2100168036 66078338996353059444JN 94 80  687-9824 1736 17K    4167FD-9976 -696    0SDSR5
D2DM0002010F45920   222 -2100168037 66978338396382359444JN164 85 9576-2746  872 19K    3160FD-2660-9425 1730SDSR5
D2DM0002010F45920   222 -2100168038 71478335396379059446JN101 87 1905-9803  523 20K    3171FD-9786-1856  870SDSR5
D2DM0002010F45920   222 -2100168049 86778325096367659451JN150 80 8529-4924 1736 32K    3159FD-3267-2439 9131SDSR5
D2DM0002010F45920   222 -2100168054 93078320896362959453JN157 85 9170-3892  872 24J    3175FD-3982-8803 2576SDSR5
D2DM0002010F45920   222 -2100168060 98278317396359159455JN207 85 8876 4523  872 10J    3170FD 4000-8508 3407SDSR5
D2DM0003010F45920   227  0 21124018    078314096363059538JN 91 83  173-9924 1219 14 I    174FD 9652  484 2569SDSR5
```

SYSTEM DOCUMENTATION

COMPUTER EQUIPMENT:

CPU:IBM 360 model 67; 32-bit word.

Memory: IBM 2365-2; 1 million bytes.

PERIPHERAL EQUIPMENT:

IBM 1403N1 printers; IBM 2501-B2 card readers; IBM 2540 card reader/punch; IBM 2301 drums; IBM 3333 and 3330 disc drives; IBM 3420-5 9-track dual-density tape drives (800/1600 bpi); IBM 2741 communications terminals; IBM 3277 display terminals; Calcomp 770/665 drum plotter (paper width 29 5/16 inches; pen speed 2.25 inches per second; increment sizes 0.0025 and 0.005 inches).

OPERATING SYSTEM:

Michigan Terminal System; a time-sharing virtual-memory system.

PROGRAM LISTING:

```
C PROGRAM KEY4          OCT 73                                          KEY   10
  1 FORMAT(A8,1X,A4,2(1X,A8),1X,A2,1X,A4,2(1X,I8),1X,A8,1X,2A1)         KEY   20
  2 FORMAT(//1X,'**** INVALID COMMAND')                                 KEY   30
  3 FORMAT(///1X,A8,1X,A4,2(1X,A8),1X,A2,1X,A4,2(1X,I8),1X,A8,1X,2A1)  KEY   40
  4 FORMAT(I2)                                                          KEY   50
  5 FORMAT(A8,26X,9A1)                                                  KEY   60
  6 FORMAT(//1X,'**** DATA NAME NOT IN TABLE')                          KEY   70
  7 FORMAT(//1X,'**** INVALID RELATIONAL OPERATOR')                     KEY   80
 15 FORMAT(//1X,'**** KEY ''',A8,''' DOES NOT EXIST')                   KEY   90
 16 FORMAT(//1X,'**** LABEL(3) ''',A8,''' IS INVALID')                  KEY  100
 17 FORMAT(//1X,'**** LABEL(2) ''',A8,''' IS INVALID')                  KEY  110
 18 FORMAT(//1X,'**** LABEL(1) ''',A8,''' IS INVALID')                  KEY  120
 19 FORMAT(//1X,'**** END OF FILE WHILE READING INPUT KEY')             KEY  130
 20 FORMAT('+',35X,'CPU TIME =',I7,' MILLISECONDS')                     KEY  140
 21 FORMAT(1X,'NEW KEY ADDED TO FILE')                                  KEY  150
    REAL*8 LABELS(40)                                                   KEY  160
    INTEGER*2 FMT(10),KEY1(5018),KEY2(5018),KEY3(5018)                  KEY  170
    COMMON/ONE/LABELS,IREADR,IPRINT,ISAVE,ITEMP,IDATA,IR1,IR2,IR3,FMT,KEY  180
   *NUMA,NUMB,LENFIL,NKEYS,LEN1,LEN2,LEN3,KEY1,KEY2,KEY3                KEY  190
    REAL*8 TITLE(15), LBL(3),BL8/'        '/,LBL1,LBL2,LBL3             KEY  200
    REAL*8 DNAME,TABLE(100)                                             KEY  210
    INTEGER READER,PRINTR,PUNCH,DATFIL                                  KEY  220
    INTEGER LENKEY(3)                                                   KEY  230
    INTEGERCODES(3,8)/1,2,1,2,1,2,2,1,1,1,2,2,1,1,2,2,2,1,1,2,2,2,2,1/KEY  240
    INTEGER OPERS(10)/'SCAN','COMM','PLUS','LESS','STOP','READ'/        KEY  250
   *,OPER,REL,RELS(11)/'EQ  ','NE  ','LT  ','NL  ','GT  ','NG  ','GE  KEY  260
   *','LE  ','IR  ','OR  ','OS  '/                                      KEY  270
   *,BL4/'    '/,ALPH                                                   KEY  280
    INTEGER*2 KEY(5018,3)                                               KEY  290
    INTEGER*2 TABLE2(9,100),AA/'A '/                                    KEY  300
    EQUIVALENCE (READER,IREADR),(PRINTR,IPRINT),(DATFIL,IDATA)          KEY  310
    EQUIVALENCE (LEN1,LENKEY),(LEN2,LENKEY(2)),(LEN3,LENKEY(3))         KEY  320
   *,(KEY3,KEY(1,3))                                                    KEY  330
    EQUIVALENCE (LBL1,LBL),(LBL2,LBL(2)),(LBL3,LBL(3))                  KEY  340
    READER=5                                                            KEY  350
    PRINTR=6                                                            KEY  360
```

```
      PUNCH=7                                                           KEY  370
      IRETR=8                                                           KEY  380
      DATFIL=1                                                          KEY  390
      ITEMP=2                                                           KEY  400
      ISAVE=7                                                           KEY  410
      NKEYS=0                                                           KEY  420
      NCUREC=1                                                          KEY  430
C          READ TABLE                                                   KEY  440
      READ(3     ,4)LENTAB                                              KEY  450
      DO 80 J=1,LENTAB                                                  KEY  460
   80 READ(3     ,5)TABLE(J),(TABLE2(I,J),I=1,9)                        KEY  470
      GOTO 90                                                           KEY  480
C          ADD NEW KEY TO LIST                                          KEY  490
   84 I=NKEYS+1                                                         KEY  500
      CALL POSN(ITEMP,NCUREC,I)                                         KEY  510
      WRITE(ITEMP)LEN3,(KEY3(K),K=1,LEN3)                               KEY  520
      NKEYS=NKEYS+1                                                     KEY  530
      LABELS(NKEYS)=LBL3                                                KEY  540
      NCUREC=NKEYS+1                                                    KEY  550
      WRITE(6,21)                                                       KEY  560
      IF(KSAVE.EQ.BL4)GOTO 86                                           KEY  570
      CALL NSAVE(LBL3)                                                  KEY  580
   86 IF(KRETR.EQ.BL4)GOTO 90                                           KEY  590
      CALL NRETR(IRETR,LBL3)                                            KEY  600
C          READ COMMAND                                                 KEY  610
   90 CONTINUE                                                          KEY  620
      READ(READER,1,END=12000)LBL1,OPER,LBL2,DNAME,REL,ALPH,NUMA,NUMB,  KEY  630
     *LBL3,KSAVE,KRETR                                                  KEY  640
      WRITE(PRINTR,3)LBL1,OPER,LBL2,DNAME,REL,ALPH,NUMA,NUMB,LBL3,KSAVE KEY  650
     *,KRETR                                                            KEY  660
C          GET KEYS                                                     KEY  670
      DO 96 I=1,2                                                       KEY  680
      IF(LBL(I).EQ.BL8)GOTO 96                                          KEY  690
      DO 92 J=1,NKEYS                                                   KEY  700
      IF(LBL(I).EQ.LABELS(J))GOTO 94                                    KEY  710
   92 CONTINUE                                                          KEY  720
```

```
      GOTO 12500                                                        KEY  730
   94 CALL RCSN(ITEMP,NCUREC,J)                                         KEY  740
      READ(ITEMP)LEN,(KEY(K,I),K=1,LEN)                                 KEY  750
      LENKEY(I)=LEN                                                     KEY  760
      NCUREC=J+1                                                        KEY  770
   96 CONTINUE                                                          KEY  780
C          GOTO APPROPRIATE PROGRAM SECTION                             KEY  790
      DO 100 I=1,10                                                     KEY  800
      IF(OPER.EQ.OPERS(I))GOTO(1000,2000,3000,4000,12000,8000),I        KEY  810
  100 CONTINUE                                                          KEY  820
      WRITE(PRINTR,2)                                                   KEY  830
      STOP12                                                            KEY  840
 1000 CONTINUE                                                          KEY  850
      REWIND DATFIL                                                     KEY  860
C          FIND NAME IN TABLE AND STORE FORMAT                          KEY  870
      IF(LBL3.EQ.BL8) GOTO 12100                                        KEY  880
      DO 1020 J=1,LENTAB                                                KEY  890
      IF(DNAME.EQ.TABLE(J)) GOTO 1040                                   KEY  900
 1020 CONTINUE                                                          KEY  910
      WRITE(PRINTR,6)                                                   KEY  920
      STOP 12                                                           KEY  930
 1040 DO 1050 I=1,9                                                     KEY  940
 1050 FMT(I)=TABLE2(I,J)                                                KEY  950
C          IDENTIFY RELATIONAL OPERATOR                                 KEY  960
      DO 1060 J=1,8                                                     KEY  970
      IF(REL.EQ.RELS(J)) GOTO 1100                                      KEY  980
 1060 CONTINUE                                                          KEY  990
      DO 1080 J=9,11                                                    KEY 1000
      IF(REL.EQ.RELS(J)) GOTO 1500                                      KEY 1010
 1080 CONTINUE                                                          KEY 1020
      WRITE(PRINTR,7)                                                   KEY 1030
      STOP 12                                                           KEY 1040
C          STORE CODES AND CALL ROUTINE                                 KEY 1050
 1100 IP1=CODES(1,J)                                                    KEY 1060
      IP2=CODES(2,J)                                                    KEY 1070
      IP3=CODES(3,J)                                                    KEY 1080
      IF(FMT(7).EQ.AA) NUMA=ALPH                                        KEY 1090
      IF(LBL2.EQ.BL8) GOTO 1150                                         KEY 1100
      CALL RREFE                                                        KEY 1110
      GOTO 84                                                           KEY 1120
 1150 CALL RFIND                                                        KEY 1130
      GOTO 84                                                           KEY 1140
C           CALL ROUTINE                                                KEY 1150
 1500 CONTINUE                                                          KEY 1160
      J=J-8                                                             KEY 1170
      IF(I2.EQ.0) GOTO(1510,1520,1530),J                                KEY 1180
      GOTO(1540,1550,1560),J                                            KEY 1190
 1510 CALL RIR                                                          KEY 1200
      GOTO 84                                                           KEY 1210
 1520 CALL RCR                                                          KEY 1220
      GOTO 84                                                           KEY 1230
 1530 CALL RCS                                                          KEY 1240
      GOTO 84                                                           KEY 1250
 1540 CALL RIRK                                                         KEY 1260
      GOTO 84                                                           KEY 1270
 1550 CALL RCRK                                                         KEY 1280
      GOTO 84                                                           KEY 1290
 1560 CALL RCSK                                                         KEY 1300
      GOTO 84                                                           KEY 1310
 2000 CONTINUE                                                          KEY 1320
      IF(LBL3.EQ.BL8) GOTO 12100                                        KEY 1330
      IF(LBL2.EQ.BL8) GOTO 12200                                        KEY 1340
      IF(LBL1.EQ.BL8) GOTO 12300                                        KEY 1350
      CALL MAND                                                         KEY 1360
      GOTO 84                                                           KEY 1370
 3000 CONTINUE                                                          KEY 1380
      IF(LBL3.EQ.BL8) GOTO 12100                                        KEY 1390
      IF(LBL2.EQ.BL8) GOTO 12200                                        KEY 1400
      IF(LBL1.EQ.BL8) GOTO 12300                                        KEY 1410
      CALL MCF                                                          KEY 1420
      GOTO 84                                                           KEY 1430
 4000 CONTINUE                                                          KEY 1440
```

```
      IF(IBL3.EQ.BL8)GOTO 12100                                         KEY 1450
      IF(LBL2.EQ.BL8)GOTO 12200                                         KEY 1460
      IF(LBL1.EQ.BL8)GOTO 4050                                          KEY 1470
      CALL MNAND                                                        KEY 1480
      GOTO 84                                                           KEY 1490
 4050 CALL MNOT                                                         KEY 1500
      GOTO 84                                                           KEY 1510
 8000 CONTINUE                                                          KEY 1520
      CALL READ(4      ,TITLE,LENFIL,LEN3,KEY3,&12600,&12600)           KEY 1530
      GOTO 84                                                           KEY 1540
12000 STOP                                                              KEY 1550
12100 WRITE(PRINTR,16)IBL3                                              KEY 1560
      STOP12                                                            KEY 1570
12200 WRITE(PRINTR,17)IBL2                                              KEY 1580
      STOP12                                                            KEY 1590
12300 WRITE(PRINTR,18)IBL1                                              KEY 1600
      STOP12                                                            KEY 1610
12500 WRITE(PRINTR,15)IBL(J)                                            KEY 1620
      STOP12                                                            KEY 1630
12600 WRITE(PRINTR,19)                                                  KEY 1640
      STOP 12                                                           KEY 1650
      END                                                               KEY 1660
      SUBROUTINE POSN(ICU,NCUREC,IPOSN)                                 KEY 1670
C POSITIONS A FILE AT A GIVEN RECORD                                     KEY 1680
      MOVE=IPOSN-NCUREC                                                 KEY 1690
      IF(MOVE)500,400,300                                               KEY 1700
  500 MOVE=IPOSN-1                                                      KEY 1710
      REWIND IOU                                                        KEY 1720
      IF(MOVE.EQ.0)RETURN                                               KEY 1730
  300 DO 350 I=1,MOVE                                                   KEY 1740
  350 READ(ICU)                                                         KEY 1750
  400 RETURN                                                            KEY 1760
      END                                                               KEY 1770
      SUBROUTINE NRETR(IOU,IBL)                                         KEY 1780
C RETRIEVES THE SUBSET OF THE DATA FILE GIVEN BY THE KEY                KEY 1790
C JUST OBTAINED                                                         KEY 1800
```

```
      REAL*8 LABELS(40)                                                 KEY 1810
      INTEGER*2 FMT(10),KEY1(5018),KEY2(5018),KEY3(5018)                KEY 1820
      COMMON/ONE/LABELS,IREADR,IPRINT,ISAVE,ITEMP,IDATA,IP1,IP2,IP3,FMT,KEY 1830
     *NUMA,NUMB,LENFIL,NKEYS,LEN1,LEN2,LEN3,KEY1,KEY2,KEY3              KEY 1840
      REAL*8 LINE(14),BLINE(14)/14*'        '/,LBL                      KEY 1850
      REWIND IDATA                                                      KEY 1860
      WRITE(IOU,2) LBL                                                  KEY 1870
      CALL RKEY(KEY3,LEN3,IDATA)                                        KEY 1880
   80 CALL NEXT(&100)                                                   KEY 1890
      READ(IDATA ,1) LINE                                               KEY 1900
      WRITE(IOU,1) LINE                                                 KEY 1910
      GOTO 80                                                           KEY 1920
  100 WRITE(IOU,1) BLINE                                                KEY 1930
      WRITE(IPRINT,3) LBL                                               KEY 1940
      RETURN                                                            KEY 1950
    1 FORMAT(14A8)                                                      KEY 1960
    2 FORMAT('T',A8)                                                    KEY 1970
    3 FORMAT(1X,A8,' HAS BEEN RETRIEVED')                               KEY 1980
      END                                                               KEY 1990
      SUBROUTINE RKEY(KEY,LENKEY,IDATA)                                 KEY 2000
C POSITIONS THE INPUT FILE ACCORDING TO A KEY.  SUCCESSIVE CALLS TO     KEY 2010
C "NEXT" POSITION THE FILE BEFORE THE NEXT RECORD SPECIFIED IN THE      KEY 2020
C KEY                                                                   KEY 2030
      INTEGER*2 KEY(5018)                                               KEY 2040
    1 FORMAT()                                                          KEY 2050
      K=0                                                               KEY 2060
      NSTOP=-1                                                          KEY 2070
      RETURN                                                            KEY 2080
      ENTRY NEXT(*)                                                     KEY 2090
      K=K+1                                                             KEY 2100
      IF(K.GT.LENKEY) RETURN1                                           KEY 2110
      NSTART=NSTOP+2                                                    KEY 2120
      NSTOP=KEY(K)-1                                                    KEY 2130
      IF(NSTART.GT.NSTOP) RETURN                                        KEY 2140
      DO 150 L=NSTART,NSTOP                                             KEY 2150
  150 READ(IDATA,1)                                                     KEY 2160
```

```
      RETURN                                                            KEY 2170
      END                                                               KEY 2180
      SUBROUTINE NSAVE(LBL)                                             KEY 2190
C WRITES OUT THE KEY JUST OBTAINED                                       KEY 2200
    1 FORMAT(1X,A8,  1X,'HAS BEEN SAVED  ')                             KEY 2210
      REAL*8 LABELS(40)                                                 KEY 2220
      INTEGER*2 FMT(10),KEY1(5018),KEY2(5018),KEY3(5018)                KEY 2230
      COMMON/ONE/LABELS,IREADR,IPRINT,ISAVE,ITEMP,IDATA,IP1,IP2,IP3,FMT,KEY 2240
     *NUMA,NUMB,LENFIL,NKEYS,LEN1,LEN2,LEN3,KEY1,KEY2,KEY3              KEY 2250
      REAL*8 TITLE(15)/15*'        '/,LBL                               KEY 2260
      INTEGER*2 KEY(5018)                                               KEY 2270
      EQUIVALENCE(KEY,KEY3),(IPUNCH,ISAVE),(LENKEY,LEN3)                KEY 2280
      TITLE(1)=LBL                                                      KEY 2290
      CALL WRITE(IPUNCH,TITLE,LENFIL,LENKEY,KEY)                        KEY 2300
      WRITE(IPRINT,1)LBL                                                KEY 2310
      RETURN                                                            KEY 2320
      END                                                               KEY 2330
      SUBROUTINE RFIND                                                  KEY 2340
C SEARCHES THE ENTIRE DATA FILE FOR THOSE RECORDS SATISFYING THE        KEY 2350
C GIVEN CONDITION                                                       KEY 2360
    4 FORMAT(//1X,'END OF DATA')                                        KEY 2370
    5 FORMAT(/1X,I5,' LINES READ    ',I5,' LINE NUMBERS IN KEY')        KEY 2380
      REAL*8 LABELS(40)                                                 KEY 2390
      INTEGER*2 FMT(10),KEY1(5018),KEY2(5018),KEY3(5018)                KEY 2400
      COMMON/ONE/LABELS,IREADR,IPRINT,ISAVE,ITEMP,IDATA,IP1,IP2,IP3,FMT,KEY 2410
     *NUMA,NUMB,LENFIL,NKEYS,LEN1,LEN2,LEN3,KEY1,KEY2,KEY3              KEY 2420
      INTEGER      VAL,VAL2,BL4/'    '/,          DATA,DATFIL,PRINTR    KEY 2430
      INTEGER*2 NEW(5018),NUM                                           KEY 2440
      EQUIVALENCE(VAL,NUMA),(VAL2,NUMB),(DATFIL,IDATA),(PRINTR,IPRINT)  KEY 2450
     *,(J,LENFIL),(NEW,KEY3),(LENGTH,LEN3)                              KEY 2460
 1000 LENGTH=0                                                          KEY 2470
      NUM=0                                                             KEY 2480
C                    FIND NEXT LINE NUMBER TO BE STORED                 KEY 2490
 1010 READ(DATFIL,FMT,END=1800)DATA                                     KEY 2500
      NUM=NUM+1                                                         KEY 2510
      IF(DATA-VAL)1015,1016,1017                                        KEY 2520
 1015 GOTO(1010,1018),IP1                                               KEY 2530
 1016 GOTO(1010,1018),IP2                                               KEY 2540
 1017 GOTO(1010,1018),IP3                                               KEY 2550
C                   STORE LINE NUMBER                                   KEY 2560
 1018 LENGTH=LENGTH+1                                                   KEY 2570
 1020 NEW(LENGTH)=NUM                                                   KEY 2580
      GOTO 1010                                                         KEY 2590
C                   END OF DATA                                         KEY 2600
 1800 J=NUM                                                             KEY 2610
      WRITE(PRINTR,4)                                                   KEY 2620
      WRITE(PRINTR,5) NUM,LENGTH                                        KEY 2630
      RETURN                                                            KEY 2640
      END                                                               KEY 2650
      SUBROUTINE RKEEP                                                  KEY 2660
C SEARCHES A SUBSET OF THE DATA FILE FOR THOSE RECORDS SATISFYING       KEY 2670
C THE GIVEN CONDITION                                                   KEY 2680
    1 FORMAT()                                                          KEY 2690
    4 FORMAT(//1X,'END OF DATA')                                        KEY 2700
    5 FORMAT(/1X,'OLD KEY CONTAINED',I5,' LINE NUMBERS',                KEY 2710
     *5X,'NEW KEY CONTAINS',I5,' LINE NUMBERS')                         KEY 2720
      REAL*8 LABELS(40)                                                 KEY 2730
      INTEGER*2 FMT(10),KEY1(5018),KEY2(5018),KEY3(5018)                KEY 2740
      COMMON/ONE/LABELS,IREADR,IPRINT,ISAVE,ITEMP,IDATA,IP1,IP2,IP3,FMT,KEY 2750
     *NUMA,NUMB,LENFIL,NKEYS,LEN1,LEN2,LEN3,KEY1,KEY2,KEY3              KEY 2760
      INTEGER       VAL,VAL2,BL4/'    '/,          DATA,DATFIL,PRINTR   KEY 2770
      INTEGER*2 NEW(5018),OLD(5018),NUM                                 KEY 2780
      INTEGER*2 M                                                       KEY 2790
      EQUIVALENCE(VAL,NUMA),(VAL2,NUMB),(DATFIL,IDATA),(PRINTR,IPRINT)  KEY 2800
     *,(NEW,KEY3),(OLD,KEY2),(NFILE,LENFIL),(LAST,LEN2),(LENGTH,LEN3)   KEY 2810
      LENGTH=0                                                          KEY 2820
      NEXT=1                                                            KEY 2830
      NUM=1                                                             KEY 2840
 3010 M=OLD(NEXT)                                                       KEY 2850
      NSKIP=M-NUM                                                       KEY 2860
      IF(NSKIP.EQ.0)GOTO 3040                                           KEY 2870
      DO 3020 I=1,NSKIP                                                 KEY 2880
 3020 READ(DATFIL,1)                                                   KEY 2890
 3040 READ(DATFIL,FMT)DATA                                             KEY 2900
      IF(DATA-VAL) 3050,3051,3052                                      KEY 2910
 3050 GOTO(3060,3055),IP1                                              KEY 2920
 3051 GOTO(3060,3055),IP2                                              KEY 2930
 3052 GOTO(3060,3055),IP3                                              KEY 2940
 3055 LENGTH=LENGTH+1                                                  KEY 2950
      NEW(LENGTH)=M                                                    KEY 2960
 3060 NUM=M+1                                                          KEY 2970
      NEXT=NEXT+1                                                      KEY 2980
      IF(NEXT-LAST)3010,3010,3800                                      KEY 2990
C                        END OF DATA                                   KEY 3000
 3800 CONTINUE                                                         KEY 3010
      WRITE(PRINTR,4)                                                  KEY 3020
      WRITE(PRINTR,5)LAST,LENGTH                                       KEY 3030
      RETURN                                                           KEY 3040
      END                                                              KEY 3050
      SUBROUTINE RIF                                                   KEY 3060
C SEARCHES THE ENTIRE DATA FILE FOR VALUES INSIDE THE GIVEN RANGE      KEY 3070
    4 FORMAT(//1X,'END OF DATA')                                       KEY 3080
    5 FORMAT(/1X,I5,' LINES READ     ',I5,' LINE NUMBERS IN KEY')      KEY 3090
      REAL*8 LABELS(40)                                                KEY 3100
      INTEGER*2 FMT(10),KEY1(5018),KEY2(5018),KEY3(5018)               KEY 3110
      COMMON/ONE/LABELS,IREADR,IPRINT,ISAVE,ITEMP,IDATA,IP1,IP2,IP3,FMT,KEY 3120
     *NUMA,NUMB,LENFIL,NKEYS,LEN1,LEN2,LEN3,KEY1,KEY2,KEY3             KEY 3130
      INTEGER      VAL,VAL2,BL4/'    '/,          DATA                 KEY 3140
      INTEGER*2 NEW(5018),OLD(5018),NUM                                KEY 3150
      EQUIVALENCE(VAL,NUMA),(VAL2,NUMB),(LENGTH,LEN3),(NEW,KEY3)       KEY 3160
     *,(OLD,KEY2),(J,LENFIL)                                           KEY 3170
 1000 LENGTH=0                                                         KEY 3180
      NUM=0                                                            KEY 3190
C                  FIND NEXT LINE NUMBER TO BE STORED                  KEY 3200
 1010 READ(IDATA,FMT,END=1800)DATA                                     KEY 3210
      NUM=NUM+1                                                        KEY 3220
      IF(DATA.GE.VAL  .AND.  DATA.LT.VAL2)GOTO 1018                    KEY 3230
      GOTO 1010                                                        KEY 3240
```

```
C                      STORE LINE NUMBER                               KEY 3250
 1018 LENGTH=LENGTH+1                                                  KEY 3260
 1020 NEW(LENGTH)=NUM                                                  KEY 3270
      GOTO 1010                                                        KEY 3280
C                      END OF DATA                                     KEY 3290
 1800 J=NUM                                                            KEY 3300
      WRITE(IPRINT,4)                                                  KEY 3310
      WRITE(IPRINT,5) NUM,LENGTH                                       KEY 3320
      RETURN                                                           KEY 3330
      END                                                              KEY 3340
      SUBROUTINE RIRK                                                  KEY 3350
C SEARCHES A SUBSET OF THE DATA FILE FOR VALUES INSIDE                 KEY 3360
C THE GIVEN RANGE                                                      KEY 3370
    1 FORMAT()                                                         KEY 3380
    4 FORMAT(//1X,'END OF DATA')                                       KEY 3390
    5 FORMAT(/1X,'OLD KEY CONTAINED',I5,' LINE NUMBERS',               KEY 3400
     *5X,'NEW KEY CONTAINS',I5,' LINE NUMBERS')                        KEY 3410
      REAL*8 LABELS(40)                                                KEY 3420
      INTEGER*2 FMT(10),KEY1(5018),KEY2(5018),KEY3(5018)               KEY 3430
      COMMON/ONE/LABELS,IREADR,IPRINT,ISAVE,ITEMP,IDATA,IP1,IP2,IP3,FMT,KEY 3440
     *NUMA,NUMB,LENFIL,NKEYS,LEN1,LEN2,LEN3,KEY1,KEY2,KEY3             KEY 3450
      INTEGER         VAL,VAL2,BL4/'    '/                             KEY 3460
      INTEGER*2 NEW(5018),OLD(5018),NUM                                KEY 3470
      INTEGER*2 M                                                      KEY 3480
      INTEGER DATA                                                     KEY 3490
      EQUIVALENCE(NFILE,LENFIL),(LAST,LEN2),(OLD,KEY2),(LENGTH,LEN3),  KEY 3500
     *(NEW,KEY3),(VAL,NUMA),(VAL2,NUMB)                                KEY 3510
      LENGTH=0                                                         KEY 3520
      NEXT=1                                                           KEY 3530
      NUM=1                                                            KEY 3540
 3010 M=OLD(NEXT)                                                      KEY 3550
      NSKIP=M-NUM                                                      KEY 3560
      IF(NSKIP.EQ.0)GOTO 3040                                          KEY 3570
      DO 3020 I=1,NSKIP                                                KEY 3580
 3020 READ(IDATA,1)                                                    KEY 3590
 3040 READ(IDATA,FMT)DATA                                              KEY 3600
```

```
      IF(DATA.GE.VAL  .AND.  DATA.LT.VAL2)GOTO 3055                    KEY 3610
      GOTO 3060                                                        KEY 3620
 3055 LENGTH=LENGTH+1                                                  KEY 3630
      NEW(LENGTH)=M                                                    KEY 3640
 3060 NUM=M+1                                                          KEY 3650
      NEXT=NEXT+1                                                      KEY 3660
      IF(NEXT-LAST)3010,3010,3800                                      KEY 3670
C                   END OF DATA                                        KEY 3680
 3800 CONTINUE                                                         KEY 3690
      WRITE(IPRINT,4)                                                  KEY 3700
      WRITE(IPRINT,5)LAST,LENGTH                                       KEY 3710
      RETURN                                                           KEY 3720
      END                                                              KEY 3730
      SUBROUTINE ROR                                                   KEY 3740
C SEARCHES THE ENTIRE DATA FILE FOR VALUES OUTSIDE                     KEY 3750
C THE GIVEN RANGE                                                      KEY 3760
    4 FORMAT(//1X,'END OF DATA')                                       KEY 3770
    5 FORMAT(/1X,I5,' LINES READ    ',I5,' LINE NUMBERS IN KEY')       KEY 3780
      REAL*8 LABELS(40)                                                KEY 3790
      INTEGER*2 FMT(10),KEY1(5018),KEY2(5018),KEY3(5018)               KEY 3800
      COMMON/ONE/LABELS,IREADR,IPRINT,ISAVE,ITEMP,IDATA,IF1,IF2,IF3,FMT,KEY 3810
     *NUMA,NUMB,LENFIL,NKEYS,LEN1,LEN2,LEN3,KEY1,KEY2,KEY3             KEY 3820
      INTEGER         VAL,VAL2,BL4/'    '/,           DATA             KEY 3830
      INTEGER*2 NEW(5018),OLD(5018),NUM                                KEY 3840
      EQUIVALENCE(LENGTH,LEN3),(VAL,NUMA),(VAL2,NUMB),(NEW,KEY3)       KEY 3850
 1000 LENGTH=0                                                         KEY 3860
      NUM=0                                                            KEY 3870
C                   FIND NEXT LINE NUMBER TO BE STORED                 KEY 3880
 1010 READ(IDATA,FMT,END=1800)DATA                                     KEY 3890
      NUM=NUM+1                                                        KEY 3900
      IF(DATA.LT.VAL  .OR.  DATA.GE.VAL2)GOTO 1018                     KEY 3910
      GOTO 1010                                                        KEY 3920
C                   STORE LINE NUMBER                                  KEY 3930
 1018 LENGTH=LENGTH+1                                                  KEY 3940
 1020 NEW(LENGTH)=NUM                                                  KEY 3950
      GOTO 1010                                                        KEY 3960
```

```
C                    END OF DATA                                        KEY 3970
 1800 J=NUM                                                             KEY 3980
      WRITE(IPRINT,4)                                                   KEY 3990
      WRITE(IPRINT,5) NUM,LENGTH                                        KEY 4000
      RETURN                                                            KEY 4010
      END                                                               KEY 4020
      SUBROUTINE ROPK                                                   KEY 4030
C SEARCHES A SUBSET OF THE DATA FILE FOR VALUES OUTSIDE THE            KEY 4040
C GIVEN RANGE                                                           KEY 4050
    1 FORMAT()                                                          KEY 4060
    4 FORMAT(//1X,'END OF DATA')                                        KEY 4070
    5 FORMAT(/1X,'OLD KEY CONTAINED',I5,' LINE NUMBERS',                KEY 4080
     *5X,'NEW KEY CONTAINS',I5,' LINE NUMBERS')                         KEY 4090
      REAL*8 LABELS(40)                                                 KEY 4100
      INTEGER*2 FMT(10),KEY1(5018),KEY2(5018),KEY3(5018)                KEY 4110
      COMMON/ONE/LABELS,IREADR,IPRINT,ISAVE,ITEMP,IDATA,IR1,IR2,IR3,FMT,KEY 4120
     *NUMA,NUMB,LENFIL,NKEYS,LEN1,LEN2,LEN3,KEY1,KEY2,KEY3              KEY 4130
      INTEGER         VAL,VAL2,BL4/'    '/                              KEY 4140
      INTEGER*2 NEW(5018),OLD(5018),NUM                                 KEY 4150
      INTEGER*2 M                                                       KEY 4160
      EQUIVALENCE(NFILE,LENFIL),(LAST,LEN2),(OLD,KEY2),(LENGTH,LEN3)    KEY 4170
     *,(NEW,KEY3),(VAL,NUMA),(VAL2,NUMB)                                KEY 4180
      LENGTH=0                                                          KEY 4190
      NEXT=1                                                            KEY 4200
      NUM=1                                                             KEY 4210
 3010 M=OLD(NEXT)                                                       KEY 4220
      NSKIP=M-NUM                                                       KEY 4230
      IF(NSKIP.EQ.0)GOTO 3040                                           KEY 4240
      DO 3020 I=1,NSKIP                                                 KEY 4250
 3020 READ(IDATA,1)                                                     KEY 4260
 3040 READ(IDATA,FMT)DATA                                               KEY 4270
      IF(DATA.LT.VAL  .OR.  DATA.GE.VAL2)GOTO 3055                      KEY 4280
      GOTO 3060                                                         KEY 4290
 3055 LENGTH=LENGTH+1                                                   KEY 4300
      NEW(LENGTH)=M                                                     KEY 4310
 3060 NUM=M+1                                                           KEY 4320
```

```
      NEXT=NEXT+1                                                       KEY 4330
      IF(NEXT-LAST) 3010,3010,3800                                      KEY 4340
C                      END OF DATA                                      KEY 4350
 3800 CONTINUE                                                          KEY 4360
      WRITE(IPRINT,4)                                                   KEY 4370
      WRITE(IPRINT,5)LAST,LENGTH                                        KEY 4380
      RETURN                                                            KEY 4390
      END                                                               KEY 4400
      SUBROUTINE ROS                                                    KEY 4410
C NOT USED IN THIS VERSION                                             KEY 4420
      REAL*8 LABELS(40)                                                 KEY 4430
      INTEGER*2 FMT(10),KEY1(5018),KEY2(5018),KEY3(5018)                KEY 4440
      COMMON/ONE/LABELS,IREADR,IPRINT,ISAVE,ITEMP,IDATA,IP1,IP2,IP3,FMT,KEY 4450
     *NUMA,NUMB,LENFIL,NKEYS,LEN1,LEN2,LEN3,KEY1,KEY2,KEY3              KEY 4460
      RETURN                                                            KEY 4470
      END                                                               KEY 4480
      SUBROUTINE ROSK                                                   KEY 4490
C NOT USED IN THIS VERSION                                              KEY 4500
      REAL*8 LABELS(40)                                                 KEY 4510
      INTEGER*2 FMT(10),KEY1(5018),KEY2(5018),KEY3(5018)                KEY 4520
      COMMON/ONE/LABELS,IREADR,IPRINT,ISAVE,ITEMP,IDATA,IP1,IP2,IP3,FMT,KEY 4530
     *NUMA,NUMB,LENFIL,NKEYS,LEN1,LEN2,LEN3,KEY1,KEY2,KEY3              KEY 4540
      RETURN                                                            KEY 4550
      END                                                               KEY 4560
      SUBROUTINE MAND                                                   KEY 4570
C PUTS INTO KEY3 THOSE NUMBERS THAT APPEAR IN BOTH KEY1                 KEY 4580
C AND KEY2                                                              KEY 4590
    1 FORMAT(/5X,'KEY SIZE',I5)                                         KEY 4600
      REAL*8 LABELS(40)                                                 KEY 4610
      INTEGER*2 FMT(10),KEY1(5018),KEY2(5018),KEY3(5018)                KEY 4620
      COMMON/ONE/LABELS,IREADR,IPRINT,ISAVE,ITEMP,IDATA,IP1,IP2,IP3,FMT,KEY 4630
     *NUMA,NUMB,LENFIL,NKEYS,LEN1,LEN2,LEN3,KEY1,KEY2,KEY3              KEY 4640
      INTEGER*2 A(5018),B(5018),C(5018),NA,NB                           KEY 4650
      EQUIVALENCE(A,KEY1),(B,KEY2),(C,KEY3),(LENA,LEN1),(LENB,LEN2)     KEY 4660
     *,(IC,LEN3)                                                        KEY 4670
      IA=0                                                              KEY 4680
      IB=0                                                              KEY 469C
      IC=0                                                              KEY 470C
  100 IA=IA+1                                                           KEY 471C
      IF(IA.GT.LENA)GOTO 600                                            KEY 472C
      NA=A(IA)                                                          KEY 473C
      IB=IB+1                                                           KEY 474C
      IF(IB.GT.LENB)GOTO 600                                            KEY 475C
      NB=B(IB)                                                          KEY 476C
  200 IF(NA-NB)300,400,500                                              KEY 477C
  300 IA=IA+1                                                           KEY 478C
      IF(IA.GT.LENA)GOTO 600                                            KEY 479C
      NA=A(IA)                                                          KEY 480C
      GOTO 200                                                          KEY 481C
  400 IC=IC+1                                                           KEY 482C
      C(IC)=NA                                                          KEY 483C
      GOTO 100                                                          KEY 484C
  500 IB=IB+1                                                           KEY 485C
      IF(IB.GT.LENB)GOTO 600                                            KEY 486C
      NB=B(IB)                                                          KEY 487C
      GOTO 200                                                          KEY 488C
  600 CONTINUE                                                          KEY 489C
      WRITE(IPRINT,1)IC                                                 KEY 490C
      RETURN                                                            KEY 491C
      END                                                               KEY 492C
      SUBROUTINE MOR                                                    KEY 493C
C PUTS INTO KEY3 THOSE NUMBERS THAT APPEAR IN EITHER                     KEY 494C
C KEY1 OR KEY2                                                          KEY 495C
    1 FORMAT(/5X,'KEY SIZE',I5)                                         KEY 496C
    5 FORMAT(//1X,'***** ERROR  --  COMBINED KEY TOO LONG; MAXIMUM 5000'KEY 497C
     */)                                                                KEY 498C
      REAL*8 LABELS(40)                                                 KEY 499C
      INTEGER*2 FMT(10),KEY1(5018),KEY2(5018),KEY3(5018)                KEY 500C
      COMMON/ONE/LABELS,IREADR,IPRINT,ISAVE,ITEMP,IDATA,IP1,IP2,IP3,FMT,KEY 501C
     *NUMA,NUMB,LENFIL,NKEYS,LEN1,LEN2,LEN3,KEY1,KEY2,KEY3              KEY 502C
      INTEGER*2 A(5018),B(5018),C(5018),NA,NB                           KEY 503C
      EQUIVALENCE(A,KEY1),(B,KEY2),(C,KEY3),(LENA,LEN1),(LENB,LEN2)     KEY 504C
     *,(IC,IFN3)                                                        KEY 5050
      IF(LENA+LENB.GT.5000) GOTO 904                                    KEY 5060
      IA=0                                                              KEY 5070
      IB=0                                                              KEY 5080
      IC=0                                                              KEY 5090
  100 IA=IA+1                                                           KEY 5100
      IF(IA.GT.LENA) GOTO 700                                           KEY 5110
      NA=A(IA)                                                          KEY 5120
  150 IB=IB+1                                                           KEY 5130
      IF(IB.GT.LENB) GOTO 800                                           KEY 5140
      NB=B(IB)                                                          KEY 5150
  200 IF(NA-NB) 300,400,500                                             KEY 5160
  300 IC=IC+1                                                           KEY 5170
      C(IC)=NA                                                          KEY 5180
      IA=IA+1                                                           KEY 5190
      IF(IA.GT.LENA) GOTO 600                                           KEY 5200
      NA=A(IA)                                                          KEY 5210
      GOTO 200                                                          KEY 5220
  400 IC=IC+1                                                           KEY 5230
      C(IC)=NA                                                          KEY 5240
      GOTO 100                                                          KEY 5250
  500 IC=IC+1                                                           KEY 5260
      C(IC)=NB                                                          KEY 5270
      GOTO 150                                                          KEY 5280
  600 IC=IC+1                                                           KEY 5290
      C(IC)=NB                                                          KEY 5300
  700 IB=IB+1                                                           KEY 5310
      IF(IB.GT.LENB) GO TO 1000                                         KEY 5320
      IC=IC+1                                                           KEY 5330
      C(IC)=B(IB)                                                       KEY 5340
      GOTO 700                                                          KEY 5350
  800 IC=IC+1                                                           KEY 5360
      C(IC)=NA                                                          KEY 5370
  900 IA=IA+1                                                           KEY 5380
      IF(IA.GT.LENA) GOTO1000                                           KEY 5390
      IC=IC+1                                                           KEY 5400
      C(IC)=A(IA)                                                       KEY 5410
      GOTO 900                                                          KEY 5420
 1000 CONTINUE                                                          KEY 5430
      WRITE(IPRINT,1)IC                                                 KEY 5440
      RETURN                                                            KEY 5450
  904 WRITE(IPRINT,5)                                                   KEY 5460
      STOP 12                                                           KEY 5470
      END                                                               KEY 5480
      SUBROUTINE MNAND                                                  KEY 5490
C PUTS INTO KEY3 THOSE NUMBERS THAT APPEAR IN KEY1 BUT NOT               KEY 5500
C IN KEY2                                                               KEY 5510
    1 FORMAT(/5X,'KEY SIZE',I5)                                         KEY 5520
      REAL*8 LABELS(40)                                                 KEY 5530
      INTEGER*2 FMT(10),KEY1(5018),KEY2(5018),KEY3(5018)                KEY 5540
      COMMON/ONE/LABELS,IREADR,IPRINT,ISAVE,ITEMP,IDATA,IP1,IP2,IP3,FMT,KEY 5550
     *NUMA,NUMB,LENFIL,NKEYS,LEN1,LEN2,LEN3,KEY1,KEY2,KEY3              KEY 5560
      INTEGER*2 A(5018),B(5018),C(5018),NA,NB                           KEY 5570
      EQUIVALENCE(A,KEY1),(B,KEY2),(C,KEY3),(LENA,LEN1),(LENB,LEN2)     KEY 5580
     *,(IC,LEN3)                                                        KEY 5590
      IA=0                                                              KEY 5600
      IB=0                                                              KEY 5610
      IC=0                                                              KEY 5620
  100 IF(IA.EQ.LENA)GOTO 500                                            KEY 5630
      IA=IA+1                                                           KEY 5640
      NA=A(IA)                                                          KEY 5650
  150 IF(IB.EQ.LENB)GOTO 400                                            KEY 5660
      IB=IB+1                                                           KEY 5670
      NB=B(IB)                                                          KEY 5680
  200 IF(NA-NB)300,100,150                                              KEY 5690
  300 IC=IC+1                                                           KEY 5700
      C(IC)=NA                                                          KEY 5710
      IF(IA.EQ.LENA)GOTO 500                                            KEY 5720
      IA=IA+1                                                           KEY 5730
      NA=A(IA)                                                          KEY 5740
      GOTO 200                                                          KEY 5750
  400 IC=IC+1                                                           KEY 5760
```

```
      C(IC)=NA                                                          KEY 5770
  450 IF(IA.EQ.LENA)GOTO 500                                            KEY 5780
      IA=IA+1                                                           KEY 5790
      IC=IC+1                                                           KEY 5800
      C(IC)=A(IA)                                                       KEY 5810
      GOTO 450                                                          KEY 5820
  500 CONTINUE                                                          KEY 5830
      WRITE(IPRINT,1)IC                                                 KEY 5840
      RETURN                                                            KEY 5850
      END                                                               KEY 5860
      SUBROUTINE MNOT                                                   KEY 5870
C NOT USED IN THIS VERSION                                              KEY 5880
      REAL*8 LABELS(40)                                                 KEY 5890
      INTEGER*2 FMT(10),KEY1(5018),KEY2(5018),KEY3(5018)                KEY 5900
      COMMON/ONE/LABELS,IREADR,IPRINT,ISAVE,ITEMP,IDATA,IE1,IE2,IE3,FMT,KEY 5910
     *NUMA,NUMB,LENFIL,NKEYS,LEN1,LEN2,LEN3,KEY1,KEY2,KEY3              KEY 5920
      RETURN                                                            KEY 5930
      END                                                               KEY 5940
      SUBROUTINE READ(ICU,TITLE,LENFIL,LENKEY,KEY,*,*)                  KEY 5950
C READS A KEY                                                           KEY 5960
    1 FORMAT(18A4,5X,I3)                                                KEY 5970
    2 FORMAT(12A4,4X,I4,4X,I4,13X,I3)                                   KEY 5980
    3 FORMAT(18I4,5X,I3)                                                KEY 5990
    4 FORMAT(//1X,'KEY CARDS OUT OF SEQUENCE')                          KEY 6000
      INTEGER TITLE(30)                                                 KEY 6010
      INTEGER*2 KEY(5018)                                               KEY 6020
      READ(ICU,1,END=200)(TITLE(I),I=1,18),N                            KEY 6030
      IF(N.NE.1)GOTO 300                                                KEY 6040
      READ(ICU,2,END=220)(TITLE(I),I=19,30),LENFIL,LENKEY,N             KEY 6050
      IF(N.NE.2)GOTO 300                                                KEY 6060
      IF(LENKEY.EQ.0)RETURN                                             KEY 6070
      KA=1                                                              KEY 6080
      KB=18                                                             KEY 6090
      NC=3                                                              KEY 6100
  100 READ(ICU,3,END=220)(KEY(I),I=KA,KB),N                             KEY 6110
      IF(N.NE.NC)GOTO 300                                               KEY 6120
      KA=KA+18                                                          KEY 613C
      IF(KA.GT.LENKEY) RETURN                                           KEY 614C
      KE=KB+18                                                          KEY 615C
      NC=NC+1                                                           KEY 616C
      GOTO 100                                                          KEY 617C
  200 RETURN1                                                           KEY 618C
  220 RETURN2                                                           KEY 619C
  300 WRITE(6,4)                                                        KEY 620C
      STOP 12                                                           KEY 621C
      END                                                               KEY 622C
      SUBROUTINE WRITE(IOU,TITLE,LENFIL,LENKEY,KEY)                     KEY 623C
C WRITES A KEY                                                          KEY 624C
    1 FORMAT(18A4,7X,'1')                                               KEY 625C
    2 FORMAT(12A4,4X,I4,4X,I4,15X,'2')                                  KEY 626C
    3 FORMAT(18I4,5X,I3)                                                KEY 627C
      INTEGER TITLE(30)                                                 KEY 628C
      INTEGER*2 KEY(5018)                                               KEY 629C
      WRITE(IOU,1) (TITLE(I),I=1,18)                                    KEY 630C
      WRITE(IOU,2) (TITLE(I),I=19,30),LENFIL,LENKEY                     KEY 631C
      IF(LENKEY.EQ.0) RETURN                                            KEY 632C
      KA=LENKEY+1                                                       KEY 633C
      KB=LENKEY+18                                                      KEY 634C
      DO 100 I=KA,KB                                                    KEY 6350
  100 KEY(I)=0                                                          KEY 636C
      KA=1                                                              KEY 637C
      KB=18                                                             KEY 638C
      N=3                                                               KEY 639C
  200 WRITE(IOU,3) (KEY(I),I=KA,KB),N                                   KEY 640C
      KA=KA+18                                                          KEY 641C
      IF(KA.GT.LENKEY) RETURN                                           KEY 642C
      KB=KB+18                                                          KEY 643C
      N=N+1                                                             KEY 644C
      GOTO 200                                                          KEY 645C
      END                                                               KEY 646C
```

VARIABLES AND SUBROUTINES:

Input variables:

LENTAB	the number of lines in the table
TABLE	names of data fields and format codes
LBL 1	label of first subset
0PER	operation code
LBL 2	label of second subset
DNAME	name of data field to be examined
REL	relational operator
ALPH	alphameric value
NUMA	first numeric value
NUMB	second numeric value
LBL 3	label to be associated with new subset
KSAVE	'save' option control character
KRETR	'retrieve' option control character

Output variables:

LBL	label associated with retrieved subset
LINE	data record
BLINE	blank line

External Subroutines Required:

None.

DATA STRUCTURES:

Structure of the input file is described above. The program does no blocking or deblocking. Present record length for the data file is 112 bytes.

Retrieved data file consists of title record, data records, and one blank record as a delimiter. This sequence is repeated if more than one subset of data is retrieved. No end-of-file marks are written in the retrieved data file.

Sets of record numbers are output as 80-byte records. Each set consists of two title records followed by records containing the numbers. All the records in each set including the two title records are numbered sequentially in columns 78-80.

STORAGE REQUIREMENTS:

Main program:		6,528
Subroutines:	PØSN	496
	RØS	264
	RIR	624
	RØSK	264
	MNØT	264
	NRETR	856
	NSAVE	520
	RIRK	800
	MAND	704
	RØR	10,680
	RKEY	616
	RFIND	752
	RKEEP	936
	RØRK	848
	MØR	1,160
	MNAND	768
	READ	952
	WRITE	904
Common blocks:	ØNE	30,512
Total storage required:		58,900 bytes

MAINTENANCE AND UPDATES:

None.

OPERATING DOCUMENTATION

OPERATOR INSTRUCTIONS:

None.

OPERATING MESSAGES:

None.

CONTROL CARDS:

```
$RUN *MØUNT
00nnnn 9TP *TAPE1* VØL = XXXXXX
00nnnn 9TP *TAPE2* VØL = XXXXXX  RING=IN
$ENDFILE
$R *FØRTG   0=ØBJ
[source program
$ENDFILE
$R  ØBJ  1=*TAPE*  2=-TEMP  3=*SØURCE*  4= *SØURCE*  5=*SØURCE*
          6=*SINK*  7=*PUNCH*  8=*TAPE2*
[table of fields,  field names and format codes
[search and retrieval instructions.
```

ERRØR RECØVERY:

No special procedures.

RUN TIME:

Cpu time required for a "scan" operation is about 1.5 seconds per 1000 records in the data file. Saving of the set of record numbers generated requires about 0.5 seconds of cpu time per 1000 record numbers in the set. The additional cpu time required to retrieve a subset of data records is about 1 second per 1000 records in the data file plus about 5 to 7.5 seconds per 1000 records retrieved.

These results were obtained using two test files containing 112-byte records, one file of about 4000 records read from a tape, and one file of about 400 records read from a disk file. In both cases the retrieved records were written out to a disk file. The figure of 7.5 seconds per 1000 records retrieved was for retrieval of the entire 4000-record file, while the smaller figure was for retrieval of the entire 400-record file.

Identification

Program name:

MAP3

Program title:

Areal-analysis package - map program.

Author:

John Ramsden

Organization:

Department of Geology, University of Alberta, Edmonton, Alberta, Canada

Date:

1 October 1974

Program version number:

3

Application

Purpose:

To generate commands for a digital incremental plotter to draw a map of survey traverses or orientations of discontinuities.

Method:

The program determines the maximum and minimum X and Y values from the data. It then determines the position of the first grid line above the maximum X value, and the position of the first grid line below the minimum X value, using the given interval between grid lines. This is repeated for the Y values. The four grid lines so determined become the borders of the map. This border is drawn, as well as all intermediate grid lines, and a title is written below the lower border. The data points are then plotted.

Optimization of plotting is achieved in the following way. Using the extreme X,Y values corresponding to the borders of the map, the program subdivides the whole rectangular region into a 60 by 60 grid, determines the number of observations within each grid square and sets up pointers to locate all observations within each grid square. This enables the program to do a raster scan on the 60 by 60 matrix to optimize plotting.

The scale of the map is determined by a scale factor read from the parameter card. This scale factor specifies the number of data units that are to correspond to one inch on the map.

Beside each data point the program writes one number and one character string of up to four characters. If the number is zero it is not plotted. From each data point a line is drawn with azimuth A and length L inches. A and L are determined from the data items NUM1 and NUM2 associated with the data point, in a way that is determined by the parameter MODE. If MODE=1 (plotting traverses) A=NUM1 and L=NUM2/SCALE (same scale as coordinates). If MODE=2 (plotting strikes) A=NUM1 and L=0.5. If MODE=3 (plotting three-dimensional directions such as poles to discontinuities) A=NUM1 and L=PROJRAD(NUM2,DPI), where PROJRAD(NUM2,DPI) is the distance on a Lambert equal-area projection with radius DPI from the centre of the projection to a point representing a plunge of NUM2 degrees. DPI is read from the parameter card.

Optional printout of the input and certain derived data is controlled by the LIST parameter. If LIST=0 no data listing is produced. If LIST=1 the input data is listed as it is read. If LIST=2, in addition to listing the input data, the program produces a second listing as the points are plotted containing the number of each data point, NUM1, NUM2, TREND, PLUNGE, PROJX and PROJY. PROJX and PROJY are the X and Y increments from the beginning to the end of the plotted line segment.

Data restrictions:

Maximum number of data points in one map is 1500.

Timing:

t=0.8s+0.02p; t=CPU time in seconds; s=number of samples (number of maps plotted); p=total number of data points in all samples.

Implementation

Original installation:

Organization:	University of Alberta Computing Services.
Computer:	IBM 360 model 67 (1024K).
I/O devices:	card, disc, tape, typewriter terminals and display terminals.
Software:	Michigan Terminal System (MTS), a virtual-storage time-sharing system.

Source language:

FORTRAN IV

Subroutines and function subprograms required:

Subroutine TPXYEQ from JR subroutine library. Subroutines PLOT WHERE PLOTS XLIMIT NUMBER LINE SYMBOL from the system plotting subroutine library.

Size:

Total program size including all required subroutines except plotting subroutines is 86488 bytes. Total size of required plotting subroutines from the plotting subroutine library at the University of Alberta Computing Services is 8752 bytes.

Requirements for I/O devices:

I/O unit no.	read/ write	Maximum FORTRAN record length (bytes)	device control statements	suggested device
1	read	2408 unformatted	none	tape; disc.
5	read	50 formatted	none	disc; card.
6	write	112 formatted	none	terminal; line printer.
7	write	102 formatted	none	terminal; line printer.
?	write	depends on plotting subroutines.	?	tape; disc.

Usage

Input:

I/O unit 1: Data.

Unformatted, as written by program INPUT1 or INPUTK1 or DSPDAT1.

I/O unit 5: Parameters.

One record containing:

columns	contents
10	MODE
11-20	Number of data units between grid lines. Right-justified integer .
21-30	Number of data units per inch. Right-justified integer or with decimal point.
31-40	DPI. Radius of projection for plotting orientations. Right-justified integer or with decimal point. If field is blank or zero, a default of 3.937 inches (20 cm.) will be used.
50	LIST. Printout parameter.

Output:

I/O unit 6: Parameter list; messages.

Input parameters are printed on this unit.

I/O unit 7: Data listing.

I/O unit ?: Output from plotting subroutines.

Unit number and device requirements depend on the plotting subroutines.

```
C
C
C AREAL-ANALYSIS PACKAGE - PROGRAM "MAP3".
C
C
      REAL A(1500),B(1500),G(1500)
      INTEGER*4 IX(1500),IY(1500),IZ(1500),TITLE(30),FMT(20)
     2 ,NUM2(1500),NUM1(1500)
      INTEGER LABEL(2,1500),BLANK/' '/
      EQUIVALENCE (A,NUM1),(B,NUM2)
      COMMON/MAP1/ LABEL,IX,IY,IZ,NUM2,NUM1,KEY,DPI ,LIST
C
C     READ PARAMETERS
C
      READ(5,103)KEY,INTGR,SCALE,DPI,LIST
  103 FORMAT(2I10,2F10.0,I10)
      WRITE(6,104)KEY,INTGR,SCALE,DPI
  104 FORMAT(1X,'MODE=',I1,5X,'DATA UNITS BETWEEN GRID LINES=',I5,5X,
     *'DATA UNITS PER INCH=',F10.4,5X,'PROJECTION RADIUS=',F7.3)
      SCALE=1.0/SCALE
      IF(DPI.EQ.0.0)DPI=3.937
      WRITE(6,105)SCALE,DPI
  105 FORMAT(1X,'INCHES PER DATA UNIT=',F10.6,10X,'PROJECTION RADIUS=',
     *F7.3)
C
C     INITIALIZE PLOT ROUTINES
C
      CALL PLOTS
      XSTART=0.0
      CALL PLOT(1.0,1.0,-3)
      CALL XLIMIT(200.0)
C     CALL FACTOR(0.5)
  150 READ(1,END=180)NN,NV,NO,NPHR,TITLE
      IF(NV.NE.6)GOTO200
      IF(NO.GT.1500)GOTO210
      WRITE(7,214)TITLE
      KA=1
      KB=100
    5 IF(KA.GT.NO)GOTO10
      IF(KB.GT.NO)KB=NO
   20 READ(1)N,IE,(IX(J),IY(J),IZ(J),LABEL(1,J),NUM1(J),NUM2(J),
     *J=KA,KB)
      IF(LIST.GT.0)WRITE(7,213)(J,IX(J),IY(J),IZ(J),LABEL(1,J),NUM1(J),N
     *UM2(J),J=KA,KB)
   30 KA=KA+100
      KB=KB+100
      GOTO5
   10 CONTINUE
      DO 160 J=1,NO
  160 LABEL(2,J)=BLANK
C
C     PLOT MAP
C
   40 CALL PPLOT(NO,INTGR,TITLE,SCALE,XSTART)
      GOTO150
C
C     TERMINATE PLOT
C
  180 CALL WHERE(X,Y,F)
```

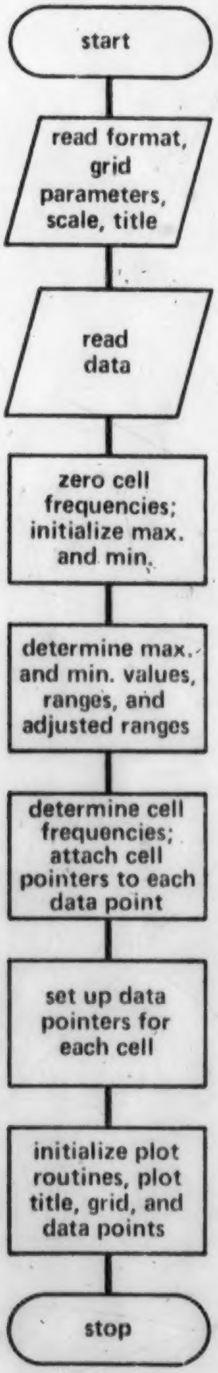
```
      CALL PLOT(X,Y,999)
      STOP
  200 WRITE(6,201)NV
      STOP12
  210 WRITE(6,211)NO
      STOP12
C
  201 FORMAT(/' **** ERROR - NUMBER OF VARIABLES IN DATA MATRIX NOT 6
     *    ****'/' NV=',I10)
  211 FORMAT(/' **** ERROR - NUMBER OF OBSERVATIONS EXCEEDS 1500 ****'/
     *' NO=',I10)
  212 FORMAT(1X,I5,2I7,I6,1X,A4,3F10.6)
  213 FORMAT(1X,I5,2I7,I6,1X,A4,2I5)
  214 FORMAT('T',30A4)
C
      END
      SUBROUTINE PPLOT( N,INTGR,TITLE,SCALE,XSTART)
C
C     DETERMINES MINIMA,MAXIMA,RANGES, MAP BORDERS  SETS UP 60X60 GRID
C      FOR OPTIMIZATION OF PLOTTING THEN CALLS ROUTINES TO DO GRID,FRAME
C     AND PLOT ALL TRAVERSE INFORMATION
C
      REAL*8 LABEL(1500)
      REAL SIZE(2)
      INTEGER*4 RANGE(2),IXMIN(2),IXMAX(2),ADDX,ADDY,IX(1500),IY(1500),
     2 IZ(1500),TITLE(1),NUM2(1500),NUM1(1500)
      INTEGER*2 L(62,62),POINT(62,62),POINT2(1500),IS(1500),JS(1500)
     2,ICODE(20)
      LOGICAL*1 LOPT,START/.TRUE./
      COMMON/MAP1/ LABEL,IX,IY,IZ,NUM2,NUM1,KEY
      COMMON/MAP2/ L,POINT,POINT2,ICODE
C
C
      IDIM=60
C
C----------ZERO CELL COUNTS
C
      DO 5 I=1,IDIM
      DO 6 J=1,IDIM
      L(I,J)=0
6     CONTINUE
5     CONTINUE
C
C----------INITIALIZE MAXIMUM AND MINIMUM
C
      IXMIN(1)=999999999
      IXMIN(2)=IXMIN(1)
      IXMAX(1)=-IXMIN(1)
      IXMAX(2)=IXMAX(1)
C
C----------DETERMINE MAXIMUM AND MINIMUM OF X AND Y
C          AND COMPUTE LIMITS OF MAP AREA
C
      DO 20 J=1,2
      DO 10 I=1,N
      IXT=IX(I)
      IF(J.EQ.2)  IXT=IY(I)
      IF(IXT.EQ.0) GO TO 10
      IF(IXT.GE.IXMIN(J)) GO TO 15
      IXMIN(J)=IXT
```

```
15      IF(IXT   .LE.IXMAX(J)) GO TO 10
        IXMAX(J)=IXT
10      CONTINUE
        IXMIN(J)=INTGR*(IXMIN(J)/INTGR)
        IXMAX(J)=INTGR*(IXMAX(J)/INTGR+1)
        RANGE(J)=IXMAX(J)-IXMIN(J)
        SIZE(J)=RANGE(J)*SCALE
20      CONTINUE
C
C----------COMPUTE DIMENSIONS OF CELLS
C
        XDELT=RANGE(1)/FLOAT(IDIM)
        YDELT=RANGE(2)/FLOAT(IDIM)
C
C----------DETERMINE CELL FREQUENCIES AND ATTACH CELL POINTERS
C----------TO EACH DATA POINT
C
        DO 25 II=1,N
        IF(IX(II).NE.0.AND.IY(II).NE.0) GO TO 26
        I=1
        J=1
        GO TO 27
26      I=(IX(II)-IXMIN(1)-1)/XDELT+1
        J=(IY(II)-IXMIN(2)-1)/YDELT+1
27      L(I,J)=L(I,J)+1
        IS(II)=I
        JS(II)=J
25      CONTINUE
C
C       SET UP POINTERS TO LOCATE SURVEY POINT INFO. FOR EACH GRID SQUARE
C
        M1=1
        DO 30 JJ=1,IDIM
        DO 31 II=1,IDIM
        IF(L(II,JJ).EQ.0) GO TO 31
        POINT(II,JJ)=M1
        M1=M1+L(II,JJ)
31      CONTINUE
30      CONTINUE
        DO 32 II=1,N
        M1=POINT(IS(II),JS(II))
        POINT2(M1)=II
        POINT(IS(II),JS(II))=POINT(IS(II),JS(II))+1
32      CONTINUE
        DO 33 II=1,N
        POINT(IS(II),JS(II))=POINT(IS(II),JS(II))-1
33      CONTINUE
C
C----------PLOT TITLE
C
   50   XP=XSTART
        YP=-1.0
        CALL SYMBOL(XP,YP,0.25,TITLE,0.0,120)
C
C----------PLOT GRID
C
        WRITE(6,1001)
 1001   FORMAT(' 1')
  100   CALL PGRID(INTGR,IXMIN,RANGE,IXMAX,SCALE,XSTART)
        WRITE(6,1002)
 1002 FORMAT(' 2')                                                      MAP31800
C                                                                       MAP31810
C----------PLOT POINTS                                                  MAP31820
C                                                                       MAP31830
      DO 40 J=1,IDIM                                                    MAP31840
      LOPT=.FALSE.                                                      MAP31850
      IF(MOD(J,2).EQ.0) LOPT=.TRUE.                                     MAP31860
      DO 41 II=1,IDIM                                                   MAP31870
      I=II                                                              MAP31880
      IF(LOPT) I=IDIM-I+1                                               MAP31890
      IF(L(I,J).EQ.0) GO TO 41                                          MAP31900
C     CALL VALID(I,J)                                                   MAP31910
      IF(KEY.EQ.1)CALL PPTS(I,J,IXMIN,SCALE,XSTART)                     MAP31920
      IF(KEY.EQ.2)CALL PPTS2(I,J,IXMIN,SCALE,XSTART)                    MAP31930
      IF(KEY.EQ.3)CALL PPTS3(I,J,IXMIN,SCALE,XSTART)                    MAP31940
41    CONTINUE                                                          MAP31950
40    CONTINUE                                                          MAP31960
C                                                                       MAP31970
C                                                                       MAP31980
      WRITE(6,1003)                                                     MAP31990
 1003 FORMAT(' 3')                                                      MAP32000
      XSTART=XSTART+SIZE(1)+3.0                                         MAP32010
  200 RETURN                                                            MAP32020
      END                                                               MAP32030
      SUBROUTINE PGRID(INTGR,IXMIN,RANGE,IXMAX,SCALE,XSTART)            MAP32040
C                                                                       MAP32050
C-----DRAWS FRAME AND GRID LINES                                        MAP32060
C                                                                       MAP32070
      REAL*4 XT(4)/0.0,0.0,0.0,1.0/,YT(4)/0.0,0.0,0.0,1.0/              MAP32080
      INTEGER*4 IXMIN(2),IXMAX(2),RANGE(2)                              MAP32090
C                                                                       MAP32100
C-----CHECK NUMBER OF GRID LINES                                        MAP32110
C                                                                       MAP32120
      DO 10 J=1,2                                                       MAP32130
      NG=(IXMAX(J)-IXMIN(J))/INTGR                                      MAP32140
      IF(NG.GT.50)GOTO40                                                MAP32150
   10 CONTINUE                                                          MAP32160
C                                                                       MAP32170
C-----PLOT GRID LINES                                                   MAP32180
C                                                                       MAP32190
      RINC=SCALE*INTGR                                                  MAP32200
      XU=XSTART+SCALE*RANGE(1)                                          MAP32210
      YU=SCALE*RANGE(2)                                                 MAP32220
      YT(1)=0.0                                                         MAP32230
      YT(2)=YU                                                          MAP32240
      XT(1)=XSTART                                                      MAP32250
      XT(2)=XSTART                                                      MAP32260
   20 CALL LINE(XT,YT,2,1,0,0)                                          MAP32270
      XT(1)=XT(1)+RINC                                                  MAP32280
      XT(2)=XT(1)                                                       MAP32290
      IF(XT(1).LE.XU)GOTO20                                             MAP32300
      XT(1)=XSTART                                                      MAP32310
      XT(2)=XU                                                          MAP32320
      YT(1)=0.0                                                         MAP32330
      YT(2)=0.0                                                         MAP32340
   30 CALL LINE(XT,YT,2,1,0,0)                                          MAP32350
      YT(1)=YT(1)+RINC                                                  MAP32360
      YT(2)=YT(1)                                                       MAP32370
      IF(YT(1).LE.YU)GOTO30                                             MAP32380
C                                                                       MAP32390
```

```
      RETURN
C
   40 WRITE(6,100)IXMAX,IXMIN,INTGR,NG
  100 FORMAT(' NUMBER OF GRID LINES IN X OR Y GREATER THAN 50 -',
     *' CHECK GRID SPACING'/1X,10I10)
      STOP12
      END
      SUBROUTINE PPTS(I,J,IXMIN,SCALE,XSTART)
C     PLOTS ALL TRAVERSE LOCATIONS, ELEVATIONS, AND NUMBERS IN GRID SQUA
      INTEGER*4 IXMIN(2),IX(1500),IY(1500),IZ(1500),LS/3/
     2,NUM2(1500),NUM1(1500)
      INTEGER*2 L(62,62),POINT(62,62),POINT2(1500),ICODE(20)
      REAL*8 LABEL(1500) ,BLANK/'        '/
      REAL*4 HTS/0.09/,HTE/.08/,HTT/0.08/,SPAC1/.03/,SPAC2/.09/,SPX1/.10
     2 /
      LOGICAL*1 LAZ
      COMMON/MAP1/ LABEL,IX,IY,IZ,NUM2,NUM1,KEY,DPI
      COMMON/MAP2/ L,POINT,POINT2,ICODE
      CONV(AZ)=3.1416/180.0*(90.0-AZ)
      CARTX(R,AZ)=R*COS(CONV(AZ))
      CARTY(R,AZ)=R*SIN(CONV(AZ))
1     N=L(I,J)
      M=POINT(I,J)
      SCALEE=SCALE
       DO 10 II=1,N
      MM=POINT2(M)
      IF(IX(MM).EQ.0 .OR. IY(MM).EQ.0)GOTO8
      AZ=FLOAT(NUM1(MM))
      LAZ=.TRUE.
      IF(AZ.LT.180.0) LAZ=.FALSE.
      XP=XSTART+SCALE*(IX(MM)-IXMIN(1))
      YP=SCALE*(IY(MM)-IXMIN(2))
      X2=XP+SPX1
      X3=X2
      IF(IZ(MM).EQ.0)GOTO15
      ELEV=FLOAT(IZ(MM))
      IF(LAZ) GO TO 20
      FLN=ALOG10(ELEV)+1.0
      X2=XP-(SPX1+6.0/7.0*FLN*HTE)
20    CALL NUMBER(X2,YP+SPAC1,HTE,ELEV,0.0,-1)
   15 IF(LABEL(MM).EQ.BLANK)GOTO9
      IF(LAZ) GO TO 30
      CALL RJUST8(LABEL(MM))
      X3=XP-(SPX1+6*HTT)
30    CALL SYMBOL(X3,YP-SPAC2,HTT,LABEL(MM),0.0,8)
9     ANG=0.0
      DO 5 JJ=1,2
      CALL SYMBOL(XP,YP,.20*JJ*HTS,1,ANG,-1)
      ANG=ANG+45.0
5     CONTINUE
C
C
      XPP=XP+CARTX(SCALEE*FLOAT(NUM2(MM)),AZ)
      YPP=YP+CARTY(SCALEE*FLOAT(NUM2(MM)),AZ)
      CALL PLOT(XPP,YPP,2)
8     M=M+1
10    CONTINUE
      RETURN
      END
      SUBROUTINE PPTS2(I,J,IXMIN,SCALE,XSTART)
```

```
C     PLOTS ALL TRAVERSE LOCATIONS, ELEVATIONS, AND NUMBERS IN GRID SQUAMAP33000
      INTEGER*4 IXMIN(2),IX(1500),IY(1500),IZ(1500),LS/3/               MAP33010
     2,NUM2(1500),NUM1(1500)                                            MAP33020
      INTEGER*2 L(62,62),POINT(62,62),POINT2(1500),ICODE(20)            MAP33030
      REAL*8 LABEL(1500) ,BLANK/'        '/                             MAP33040
      REAL*4 HTS/0.09/,HTE/.08/,HTT/0.08/,SPAC1/.03/,SPAC2/.09/,SPX1/.10MAP33050
     2 /                                                                MAP33060
      LOGICAL*1 LAZ                                                     MAP33070
      COMMON/MAP1/ LABEL,IX,IY,IZ,NUM2,NUM1                             MAP33080
      COMMON/MAP2/ L,POINT,POINT2,ICODE                                 MAP33090
      CONV(AZ)=3.1416/180.0*(90.0-AZ)                                   MAP33100
      CARTX(R,AZ)=R*COS(CONV(AZ))                                       MAP33110
      CARTY(R,AZ)=R*SIN(CONV(AZ))                                       MAP33120
1     N=L(I,J)                                                          MAP33130
      M=POINT(I,J)                                                      MAP33140
       DO 10 II=1,N                                                     MAP33150
      MM=POINT2(M)                                                      MAP33160
      IF(IX(MM).EQ.0 .OR. IY(MM).EQ.0)GOTO8                             MAP33170
      AZ=FLOAT(NUM1(MM))                                                MAP33180
      LAZ=.TRUE.                                                        MAP33190
      IF(AZ.LT.180.0) LAZ=.FALSE.                                       MAP33200
      XP=XSTART+SCALE*(IX(MM)-IXMIN(1))                                 MAP33210
      YP=SCALE*(IY(MM)-IXMIN(2))                                        MAP33220
      X2=XP+SPX1                                                        MAP33230
      X3=X2                                                             MAP33240
      IF(IZ(MM).EQ.0)GOTO15                                             MAP33250
      ELEV=FLOAT(IZ(MM))                                                MAP33260
      IF(LAZ) GO TO 20                                                  MAP33270
      FLN=ALOG10(ELEV)+1.0                                              MAP33280
      X2=XP-(SPX1+6.0/7.0*FLN*HTE)                                      MAP33290
20    CALL NUMBER(X2,YP+SPAC1,HTE,ELEV,0.0,-1)                          MAP33300
   15 IF(LABEL(MM).EQ.BLANK)GOTO9                                       MAP33310
      IF(LAZ) GO TO 30                                                  MAP33320
      CALL RJUST8(LABEL(MM))                                            MAP33330
      X3=XP-(SPX1+6*HTT)                                                MAP33340
   30 CONTINUE                                                          MAP33350
9     ANG=0.0                                                           MAP33360
      DO 5 JJ=1,2                                                       MAP33370
      CALL SYMBOL(XP,YP,.20*JJ*HTS,1,ANG,-1)                            MAP33380
      ANG=ANG+45.0                                                      MAP33390
5     CONTINUE                                                          MAP33400
      XPP=XP+CARTX(0.5,AZ)                                              MAP33410
      YPP=YP+CARTY(0.5,AZ)                                              MAP33420
      CALL PLOT(XPP,YPP,2)                                              MAP33430
8     M=M+1                                                             MAP33440
10    CONTINUE                                                          MAP33450
      RETURN                                                            MAP33460
      END                                                               MAP33470
      SUBROUTINE PPTS3(I,J,IXMIN,SCALE,XSTART)                          MAP33480
C     PLOTS ALL TRAVERSE LOCATIONS, ELEVATIONS, AND NUMBERS IN GRID SQUAMAP33490
      INTEGER*4 IXMIN(2),IX(1500),IY(1500),IZ(1500),LS/3/               MAP33500
     2,NUM2(1500),NUM1(1500)                                            MAP33510
      INTEGER*2 L(62,62),POINT(62,62),POINT2(1500),ICODE(20)            MAP33520
      REAL*8 LABEL(1500) ,BLANK/'        '/                             MAP33530
      REAL*4 HTS/0.09/,HTE/.08/,HTT/0.08/,SPAC1/.03/,SPAC2/.09/,SPX1/.10MAP33540
     2 /                                                                MAP33550
      LOGICAL*1 LAZ                                                     MAP33560
      COMMON/MAP1/ LABEL,IX,IY,IZ,NUM2,NUM1,KEY,DPI ,LIST               MAP33570
      COMMON/MAP2/ L,POINT,POINT2,ICODE                                 MAP33580
      REAL DTR/1.745329E-2/,TP(2),XY(2)                                 MAP33590
      EQUIVALENCE (TREND,TP),(PLUNGE,TP(2)),(PROJX,XY),(PROJY,XY(2))     MAP33600
1     N=L(I,J)                                                           MAP33610
      M=POINT(I,J)                                                       MAP33620
       DO 10 II=1,N                                                      MAP33630
      MM=POINT2(M)                                                       MAP33640
      IF(IX(MM).EQ.0 .OR. IY(MM).EQ.0)GOTO8                              MAP33650
      AZ=FLOAT(NUM1(MM))                                                 MAP33660
      LAZ=.TRUE.                                                         MAP33670
      IF(AZ.LT.180.0) LAZ=.FALSE.                                        MAP33680
      XP=XSTART+SCALE*(IX(MM)-IXMIN(1))                                  MAP33690
      YP=SCALE*(IY(MM)-IXMIN(2))                                         MAP33700
      X2=XP+SPX1                                                         MAP33710
      X3=X2                                                              MAP33720
      IF(IZ(MM).EQ.0)GOTO15                                              MAP33730
      ELEV=FLOAT(IZ(MM))                                                 MAP33740
      IF(LAZ) GO TO 20                                                   MAP33750
      FLN=ALOG10(ELEV)+1.0                                               MAP33760
      X2=XP-(SPX1+6.0/7.0*FLN*HTE)                                       MAP33770
20    CALL NUMBER(X2,YP+SPAC1,HTE,ELEV,0.0,-1)                           MAP33780
   15 IF(LABEL(MM).EQ.BLANK)GOTO9                                        MAP33790
      CALL LEN1(LABEL(MM),8,1,NC)                                        MAP33800
      IF(LAZ) GO TO 30                                                   MAP33810
      X3=XP-(SPX1+NC*6.0/7.0*HTT)                                        MAP33820
   30 CALL SYMBOL(X3,YP-SPAC2,HTT,LABEL(MM),0.0,NC)                      MAP33830
9     ANG=0.0                                                            MAP33840
      DO 5 JJ=1,2                                                        MAP33850
      CALL SYMBOL(XP,YP,.20*JJ*HTS,1,ANG,-1)                             MAP33860
      ANG=ANG+45.0                                                       MAP33870
5     CONTINUE                                                           MAP33880
      IF(LIST.EQ.2)WRITE(7,103)MM,NUM1(MM),NUM2(MM)                      MAP33890
  103 FORMAT(1X,I4,5X,'NUM1=',I4,5X,'NUM2=',I4)                          MAP33900
      TREND=AZ                                                           MAP33910
      PLUNGE=NUM2(MM)                                                    MAP33920
      IF(LIST.EQ.2)WRITE(7,101)TREND,PLUNGE                              MAP33930
  101 FORMAT('+',37X,'TREND=',F6.1,5X,'PLUNGE=',F6.1)                    MAP33940
      PLUNGE=PLUNGE*DTR                                                  MAP33950
      TREND=TREND*DTR                                                    MAP33960
  100 CALL TPXYEQ(TREND,PLUNGE,1,DPI,PROJX,PROJY,1,1,IER)                MAP33970
      IF(IER.NE.0)STOP13                                                 MAP33980
      IF(LIST.EQ.2)WRITE(7,102)PROJX,PROJY                               MAP33990
  102 FORMAT('+',72X,'PROJX=',F6.3,5X,'PROJY=',F6.3)                     MAP34000
      XPP=XP+PROJX                                                       MAP34010
      YPP=YP+PROJY                                                       MAP34020
      CALL PLOT(XP,YP,3)                                                 MAP34030
      CALL PLOT(XPP,YPP,2)                                               MAP34040
8     M=M+1                                                              MAP34050
10    CONTINUE                                                           MAP34060
      RETURN                                                             MAP34070
      END                                                                MAP34080
      SUBROUTINE RJUST8(A)                                               MAP34090
C RIGHT-JUSTIFIES NON-BLANK CHARACTERS IN AN 8-CHARACTER STRING          MAP34100
      INTEGER*2 C,BL/'  '/                                               MAP34110
      LOGICAL*1 A(8),B(2)/' ',' '/,BL1/' '/                              MAP34120
      EQUIVALENCE(B,C)                                                   MAP34130
C                                                                        MAP34140
      L=8                                                                MAP34150
   50 B(1)=A(L)                                                          MAP34160
      IF(C.NE.BL)GOTO100                                                 MAP34170
      L=L-1                                                              MAP34180
      IF(L)140,140,50                                                    MAP34190
```

```
C
  100 IF(L.EQ.8) RETURN
      K=8
  110 A(K)=A(L)
      K=K-1
      L=L-1
      IF(L)120,120,110
  120 A(K)=BL1
      K=K-1
      IF(K)140,140,120
  140 RETURN
      END
C
C
C SUBROUTINE:  LEN1
C
C PURPOSE:  TO DETERMINE THE NUMBER OF CHARACTERS IN A CHARACTER STRING
C           UP TO AND INCLUDING THE RIGHTMOST NON-BLANK CHARACTER.
C
C PARAMETERS:
C S LOCATION OF STRING
C L NUMBER OF CHARACTERS IN STRING
C K SPACING OF CHARACTERS
C N RETURNED NUMBER OF CHARACTERS UP TO AND INCLUDING THE
C   RIGHTMOST NON-BLANK CHARACTER
C
C SUBROUTINES CALLED:  NONE.
C
C
      SUBROUTINE LEN1(S,L,K,N)
      LOGICAL*1 S(L),B(2)
      INTEGER*2 C,BLANK/'  '/
      EQUIVALENCE (B,C)
C
      C=BLANK
      N=L
      I=(L-1)*K+1
    1 B(1)=S(I)
      IF(C.NE.BLANK)GOTO2
      N=N-1
      I=I-K
      IF(I)2,2,1
    2 RETURN
      END
```


start
read format, grid parameters, scale, title
read data
zero cell frequencies; initialize max. and min.
determine max. and min. values, ranges, and adjusted ranges
determine cell frequencies; attach cell pointers to each data point
set up data pointers for each cell
initialize plot routines, plot title, grid, and data points
stop

Flow Diagram for Program MAP3

Identification

Program name:

MODEL1

Program title:

Areal-analysis package - statistics for weighted points on the hemisphere.

Author:

John Ramsden

Organization:

Department of Geology, University of Alberta, Edmonton, Alberta, Canada

Date:

1 October 1974

Program version number:

1

Application

Purpose:

To compute "Bingham" statistics and "Fisher" statistics for weighted points on the hemisphere.

Method:

The program calls the subroutine DCPM1 to compute the data cross-product matrix, then subroutine ESTAT1 to estimate population parameters using Bingham's model. This subroutine also tests the hypotheses of uniformity and axial symmetry using the eigenvalues of the data cross-product matrix. Depending on the results of these tests, the program performs a more sensitive test for axial symmetry, using Kuiper's statistic, by calling the subroutine CIRC1.

"Fisher" statistics are obtained by calling subroutine SCOM1 to compute sums of components, subroutine FSTAT1 to estimate population parameters using Fisher's model, and subroutine CIRC1 to test for axial symmetry about the Fisher mean.

Data restrictions:

Maximum number of data points in one sample is 100.

Timing:

The program requires about 2 seconds of CPU time to process one sample containing 300 points.

Implementation

Original installation:

Organization:	University of Alberta Computing Services.
Computer:	IBM 360 model 67 (1024K).
I/O devices:	card, disc, tape, typewriter terminals and display terminals.
Software:	Michigan Terminal System (MTS), a virtual-storage time-sharing system.

Source language:

FORTRAN IV

Subroutines and function subprograms required

Subroutines DCPM1 ESTAT1 CIRC1 SCOM1 FSTAT1 BSTATG VRM1 RTSP ANDC DCAN XKF BSTATC KUIP2 SYMM ROTMAT and functions XK1 XK3 and TAB1 from JR subroutine library. Subroutines GTPRD GMPRD and EIGEN from SSP subroutine library.

Size:

Total program size including all required subroutines is 86184 bytes.

Requirements for I/O devices:

I/O unit no.	read/ write	Maximum FORTRAN record length (bytes)	device control statements	suggested device
1	read	1600 unformatted	none	tape; disc.
6	write	76 formatted	none	line printer
7	write	30 formatted	none	disc; card punch

Usage

Input:

I/O unit 1: input data.

Unformatted records, each record containing direction cosines and weights for up to 100 points. The sequence "direction cosines, weight" (four fullword real numbers) is repeated for each data point. As written by program INPUT1 or INPUTK1.

Output:

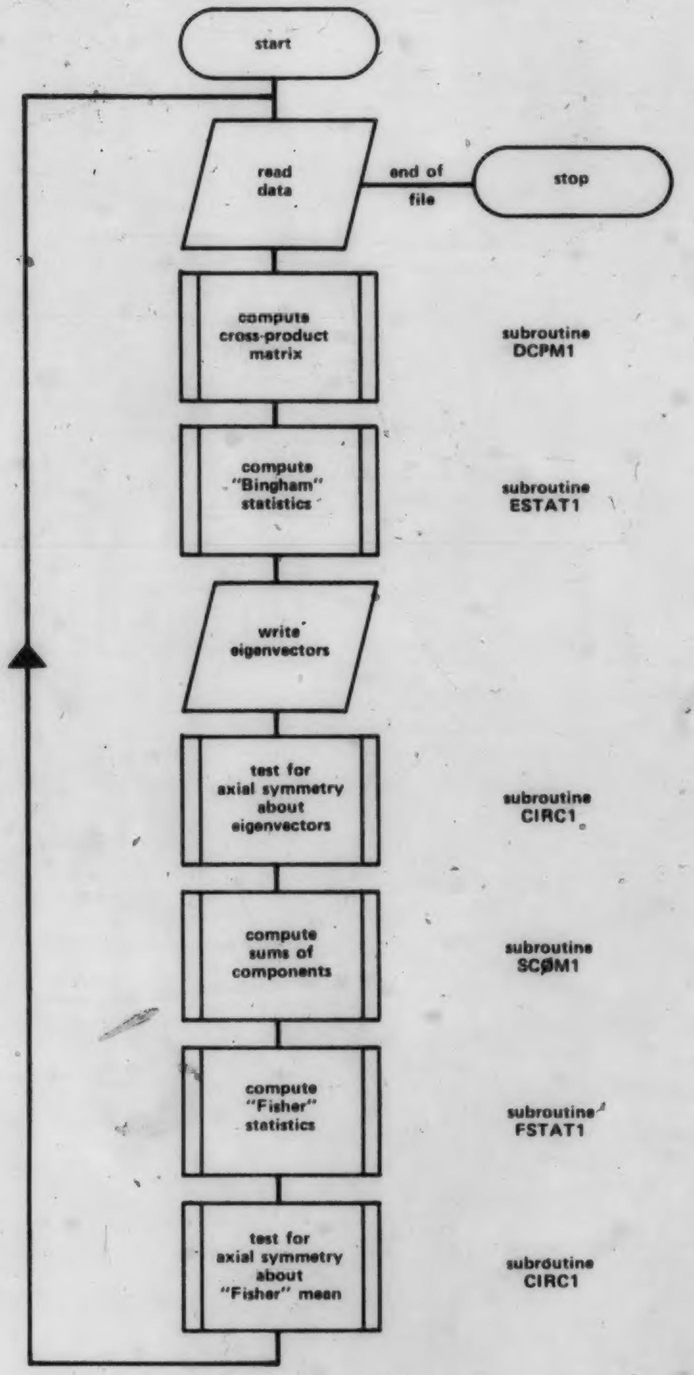
I/O unit 6: printout of statistics.

All computed statistics are written on this I/O unit, with printer carriage control.

I/O unit 7: eigenvectors.

The eigenvectors of the data cross-product matrix are written on this unit for later use by other programs. Three formatted records are written, each record containing the direction cosines of one eigenvector in the normal sequence (north, east, down). The sequence of eigenvectors is such that the associated eigenvalues are in ascending order.

```fortran
C
C
C AREAL-ANALYSIS PACKAGE - PROGRAM "MODEL1".
C
C
      REAL DC(3,2000),W(2000),U(6),E(3,3),S(12),T(6),R(4)
      REAL DCR(3,2000)
      INTEGER TITLE(30)
C
C
    1 FORMAT(' '/' '/' TEST OF UNIFORMITY ABOUT EIGENVECTOR',I2)
    2 FORMAT(' '/' '/' TEST OF UNIFORMITY ABOUT FISHER MEAN')
    3 FORMAT(3F10.6)
    6 FORMAT('1')
    5 FORMAT(//////)
    4 FORMAT(1X,F10.0)
    7 FORMAT(1X,I5,1X,30A4)
    8 FORMAT(' *** NOT PROCESSED')
    9 FORMAT(1X,I4,4F10.6)
C
C
      WEIGHT=-1.0
      READ(5,4,END=80)WEIGHT
C
C
   80 READ(1,END=900)NN,NV,NO,NPHR,TITLE
      WRITE(9,7)NO,TITLE
      IF(NO.LT.10)GOTO400
      KA=1
      KB=100
   85 IF(KA.GT.NO)GOTO90
      IF(KB.GT.NO)KB=NO
      READ(1)N,NE,((DC(I,J),I=1,3),W(J),J=KA,KB)
      KA=KA+100
      KB=KB+100
      GOTO85
   90 SW=0.0
      DO 95 J=1,NO
      IF(WEIGHT)95,94,93
   93 IF(W(J).GT.WEIGHT)W(J)=WEIGHT
      GOTO95
   94 W(J)=1.0
   95 SW=SW+W(J)
      DO 97 J=1,NO
   97 WRITE(6,9)J,(DC(I,J),I=1,3),W(J)
      N=NO
      CALL DCPM1(DC,W,N,U)
      WRITE(6,6)
 1100 CALL ESTAT1(N,TITLE,SW,U,E,S,T,IR,XK,R,6)
      WRITE(7,3) ((E(I,J),J=1,3),I=1,3)
      IR=IR+1
      GOTO(100,200,210,300),IR
  100 DO 180 J=1,3
      WRITE(6,1)J
      CALL CIRC1(DC,DCR,W,N,SW,E(1,J),3,IR,6)
  180 CONTINUE
      GOTO300
  200 J=1
      GOTO220
  210 J=3
  220 WRITE(6,1)J
      CALL CIRC1(DC,DCR,W,N,SW,E(1,J),3,IR,6)
  300 CONTINUE
C
      CALL SCOM1(DC,W,N,E(1,3),3,U)
      WRITE(6,5)
      CALL FSTAT1(TITLE,N,SW,U,E,S,T,XK,R,6)
      WRITE(6,2)
      CALL CIRC1(DC,DCR,W,N,SW,E,3,IR,6)
      GOTO80
C
  400 WRITE(9,8)
      READ(1)
      GOTO80
C
  900 STOP
      END
```


start
read data
end of file
stop
compute cross-product matrix
subroutine DCPM1
compute "Bingham" statistics
subroutine ESTAT1
write eigenvectors
test for axial symmetry about eigenvectors
subroutine CIRC1
compute sums of components
subroutine SCØM1
compute "Fisher" statistics
subroutine FSTAT1
test for axial symmetry about "Fisher" mean
subroutine CIRC1

FLOW DIAGRAM FOR PROGRAM MODEL1

Identification

Program name:

NPPLOT1

Program title:

Areal-analysis package - calcomp point diagram.

Author:

John Ramsden

Organization:

Department of Geology, University of Alberta, Edmonton, Alberta, Canada

Date:

1 October 1974

Program version number:

1

Application

Purpose:

To generate commands for a digital incremental plotter to produce a point diagram of non-polar orientations on Lambert's equal-area projection.

Method:

The orientations must be given as direction cosines. These are converted into X,Y coordinates on the projection by the subroutine RTXYEQ. The projection is horizontal unless defined otherwise by a parameter card. If a non-horizontal projection is requested, the data are rotated before the subroutine RTXYEQ is called.

Data restrictions:

Maximum number of points in any plot is 1000.

Timing:

0.006p; t=CPU time in seconds; s=number of samples (number of plots produced); p=total number of data points in all plots.

Implementation

Original installation:

Organization:	University of Alberta Computing Services.
Computer:	IBM 360 model 67 (1024K).
I/O devices:	card, disc, tape, typewriter terminals and display terminals.
Software:	Michigan Terminal System (MTS), a virtual-storage time-sharing system.

Source language:

FORTRAN IV

Subroutines and function subprograms required:

The following subroutines from JR subroutine library: VRM1 ROTMAT DCAN RTXYEQ. The following subroutines from SSP subroutine library: GMPRD GTPRD. The following subroutines from the system plotting subroutine library: PLOTS SYMBOL CIRCLE NUMBER PLOT.

Size:

Size including all required subroutines except plotting subroutines is 45696 bytes. Size of required plotting subroutines at original installation is 7920 bytes.

Requirements for I/O devices:

I/O unit no.	read/ write	Maximum FORTRAN record length (bytes)	device control statements	suggested device
1	read	1608 unformatted	none	tape; disc.
4	read	determined by given format	none	disc; card.
5	read	80 formatted	none	disc; card.
6	write	81 formatted	none	line printer
?	write	depends on plotting subroutines	?	tape; disc.

Usage

Input:

I/O unit 1: data

As written by programs INPUT1, INPUTK1, or DSPDAT1. As many samples as desired may follow sequentially. One plot will be produced for each sample.

I/O unit 4: normal to projection plane

Used only if a "P" or "Q" parameter card is supplied with a non-positive number in the fourth numeric field. Format given on parameter card.

I/O unit 5: parameters

Parameter cards are optional. The reading of parameter cards must be terminated by a blank card or an end-of-file. The parameter cards are:

(1) Normal to projection plane. Default: vertical. Card format:

col	contents
1	"P"
2-11	trend or first direction cosine
12-21	plunge or second direction cosine
22-31	blank or third direction cosine
32-40	code: +-1.0 if trend and plunge given in degrees +-2.0 if trend and plunge given in radians +-3.0 if direction cosines given code>0 if information is on parameter card code<0 if information to be read on unit 4
41-80	format for reading unit 4 if required

If this parameter card is supplied with "Q" in column 1 two plots will be produced, one using the horizontal projection plane and one using the specified projection plane.

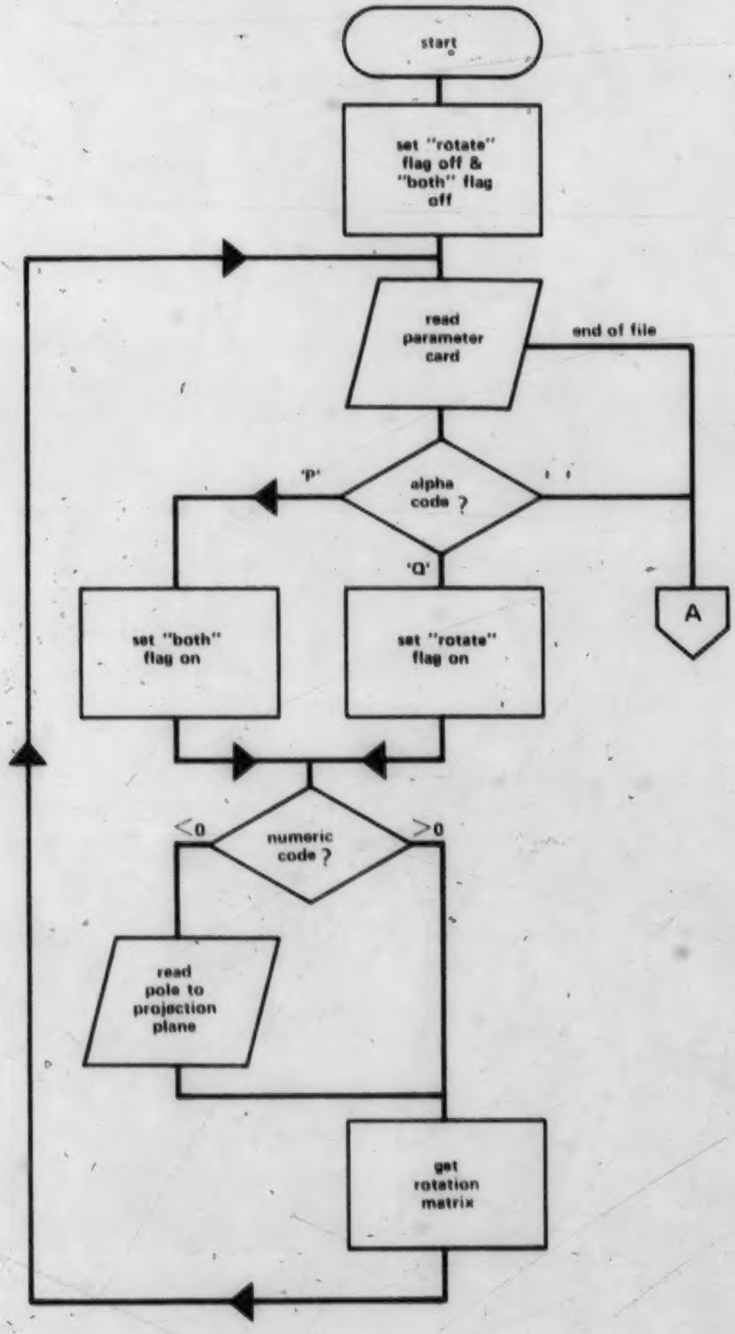
Output:

I/O unit 6: error messages

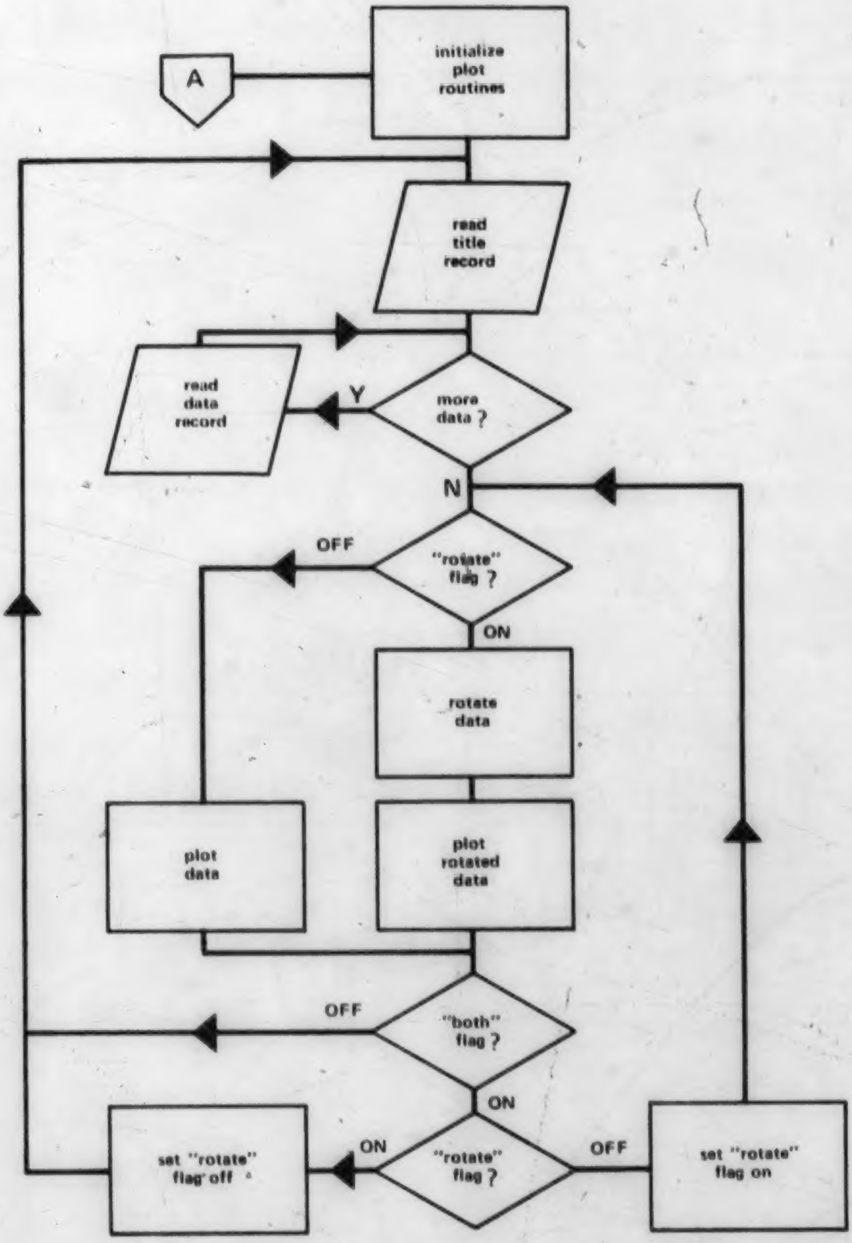
I/O unit ?: output from plotting subroutines

Unit number and device requirements depend on plotting subroutines.

```
C
C
C AREAL-ANALYSIS PACKAGE - PROGRAM "NPPLOT1".
C
C
      REAL DC(3,1000),W(1000),DCR(3,1000)
      INTEGER TITLE(30)
      LOGICAL*1 ROTAT/.FALSE./,BOTH/.FALSE./,ALPHA(40)
      REAL PPP/'P'/,BLA/' '/,X(4),RM(9),KEW/'Q'/
C
    1 FORMAT(A1,3F10.0,F9.0,40A1)
    2 FORMAT(/1X,'INVALID PARAMETER CARD'/1X,A1,4F10.4/
     *1X,'"',A1,'" IS INVALID')
    3 FORMAT(/1X,'INVALID PARAMETER CARD'/1X,A1,4F10.4/
     *1X,'"',F10.4,'" IS INVALID')
C
   15 READ(5,1,END=50)A,X,ALPHA
      IF(A.EQ.PPP)GOTO19
      IF(A.EQ.KEW)GOTO20
      IF(A.EQ.BLA)GOTO50
      WRITE(6,2)A,X,A
      STOP12
   19 BOTH=.TRUE.
      GOTO22
   20 ROTAT=.TRUE.
   22 J=X(4)
      IF(J.NE.0 .AND. IABS(J).LT.4)GOTO25
      WRITE(6,3)A,X,X(4)
      STOP12
   25 IF(J.GT.0)GOTO30
      J=-J
      GOTO(26,26,27),J
   26 READ(4,ALPHA)X(1),X(2)
      GOTO30
   27 READ(4,ALPHA)X(1),X(2),X(3)
      GOTO30
   30 CALL VRM1(X,RM,J)
      GOTO15
   50 CALL PLOTS
      IPLOT=0
C
C     READ SAMPLE
C
   90 READ(1,END=300)NN,NV,NO,TITLE
      KA=1
      KB=100
   94 IF(KA.GT.NO)GOTO98
      IF(KB.GT.NO)KB=NO
      READ(1)((DC(I,J),I=1,3),W(J),J=KA,KB)
      KA=KA+100
      KB=KB+100
      GOTO94
   98 CONTINUE
      N=NO
  100 CONTINUE
      IPLOT=IPLOT+1
      YP=(MOD(IPLOT,3)-1)*9.0+4.5
      XP=(IPLOT/3)*12.0+5.5
      IF(.NOT.ROTAT)GOTO200
      CALL GMPRD(RM,DC,DCR,3,3,N)
  180 CALL CC1(0,N,DCR,TITLE,120,XP,YP,3.937)
      GOTO250
  200 CONTINUE
  225 CALL CC1(0,N,DC,TITLE,120,XP,YP,3.937)
  250 IF(.NOT.BOTH)GOTO90
      IF(ROTAT)GOTO260
      ROTAT=.TRUE.
      GOTO100
  260 ROTAT=.FALSE.
      GOTO90
  300 CALL PLOT(0.0,0.0,999)
      STOP
      END
      SUBROUTINE CC1(J,N,DATA,ID,NCH,XP,YP,RA)
C CALLS PLOTTING SUBROUTINES TO PLOT THE POINT DIAGRAM
      REAL DATA(3000)
      REAL XY(2000)
      LOGICAL*1 ID(1)
C
    1 FORMAT(//1X,'SUBROUTINE "CC1": ERROR CODE',I3,' FROM SUBROUTINE "R
     *TXYEQ".  EXECUTION TERMINATED.')
C
      CALL RTXYEQ(DATA(1),DATA(2),DATA(3),3,RA,XY(1),XY(2),2,N,IER)
      IF(IER.NE.0)GOTO500
      TRA=RA/10.0
      TRA2=TRA/2.0
      HW=TRA/3.5
      XX=XP-RA-TRA
      YY=YP-HW
      CALL SYMBOL(XX,YY,TRA,85,90.0,-1)
      CALL SYMBOL(XP-RA,YP,TRA,13,90.0,-1)
      CALL CIRCLE(XP-RA,YP,180.0,90.0,RA,RA,0.0)
      CALL SYMBOL(XP,YP+RA,TRA,13,0.0,-1)
      CALL CIRCLE(XP,YP+RA,90.0,0.0,RA,RA,0.0)
      CALL SYMBOL(XP+RA,YP,TRA,13,90.0,-1)
      CALL CIRCLE(XP+RA,YP,0.0,-90.0,RA,RA,0.0)
      CALL SYMBOL(XP,YP-RA,TRA,13,0.0,-1)
      CALL CIRCLE(XP,YP-RA,-90.0,-180.0,RA,RA,0.0)
      CALL SYMBOL(XP,YP,TRA,3,0.0,-1)
      CS=RA/20.0
      CH=CS
      XX=XP+RA+CH+TRA
      YY=YP-CS-CS
      CALL SYMBOL(XX,YY,CH,85,90.0,-1)
      YY=YP-CS
      CALL SYMBOL(XX,YY,CH,126,90.0,-1)
      AN=N
      CALL NUMBER(XX,YP,CH,AN,90.0,-1)
      NC=NCH
      IF(NC.EQ.0)GOTO250
      IF(NC.GT.120)NC=120
      ISTART=-39
      IFINIS=0
  200 ISTART=ISTART+40
      IF(ISTART.GT.NC)GOTO250
      IFINIS=IFINIS+40
      IF(IFINIS.GT.NC)IFINIS=NC
      NCHAR=IFINIS-ISTART+1
      XX=XX+CH+CH
      YY=YP-RA                                                          NPPL1200
      CALL SYMBOL(XX,YY,CH,ID(ISTART),90.0,NCHAR)                       NPPL1210
      GOTO200                                                           NPPL1220
  250 K=0                                                               NPPL1230
      DO 300 I=1,N                                                      NPPL1240
      K=K+2                                                             NPPL1250
      XX=XP-XY(K)                                                       NPPL1260
      YY=YP+XY(K-1)                                                     NPPL1270
      CALL SYMBOL(XX,YY,0.05,1,0.0,-1)                                  NPPL1280
  300 CONTINUE                                                          NPPL1290
      RETURN                                                            NPPL1300
  500 WRITE(6,1)IER                                                     NPPL1310
      STOP12                                                            NPPL1320
      END                                                               NPPL1330
```


start
set "rotate" flag off & "both" flag off
read parameter card
end of file
alpha code ?
'P'
' '
'Q'
A
set "both" flag on
set "rotate" flag on
numeric code ?
<0
>0
read pole to projection plane
get rotation matrix

FLOW DIAGRAM FOR PROGRAM NPPLOT1 - PART 1


A
initialize plot routines
read title record
more data ?
Y
N
read data record
OFF
"rotate" flag ?
ON
rotate data
plot data
plot rotated data
OFF
"both" flag ?
ON
ON
"rotate" flag ?
OFF
set "rotate" flag off
set "rotate" flag on

FLOW DIAGRAM FOR PROGRAM NPPLOT1 - PART 2

Identification

Program name:

ROSE

Program title:

Smoothed circular histograms.

Author:

John Ramsden

Organization:

Department of Geology, University of Alberta, Edmonton, Alberta, Canada

Date:

1 October 1974

Program version number:

1

Application

Purpose:

To produce on the Calcomp plotter smoothed circular histograms (rose diagrams) given a vector of relative frequencies in 360 1-degree intervals.

Method:

The given density function (vector of relative frequencies) is smoothed using as a smoothing function the von Mises probability density function with specified parameter.

Data restrictions:

The given density function must be normalized, that is, it must consist of 360 relative frequencies whose sum is unity.

Timing:

6.3 seconds CPU time required to process 3 samples, producing 1 smoothed plot for each sample using a smoothing function parameter of 50.

Implementation

Original installation:

Organization:	University of Alberta Computing Services.
Computer:	IBM 360 model 67 (1024K).
I/O devices:	card, disc, tape, typewriter terminals and display terminals.
Software:	Michigan Terminal System (MTS), a virtual-storage time-sharing system.

Source language:

FORTRAN IV

Subroutines required:

Subroutines from system plotting subroutine library: PLOTS, PLOT, SYMBOL, NUMBER, POLAR, WHERE.

Size:

Total size of user-supplied program = 8712 bytes. Total size of required library subroutines at original installation = 14608 bytes. Total 23320 bytes.

Requirements for I/O devices:

I/O unit no.	read/ write	Maximum FORTRAN record length	device control statements	suggested device
1	read	1564 unformatted	none	tape, disc
5	read	80 formatted	none	card reader
6	write	126 formatted	none	line printer
?	write	depends on plotting subroutines	?	tape, disc

Usage

Input:

I/O unit 1: data.

Any number of unformatted records, each consisting of: 120 bytes containing a title, 4 bytes containing the number of observations as a fullword integer, and 1440 bytes containing the relative frequencies as 360 one word real numbers.

I/O unit 5: parameters.

Three parameter records. The first contains the length of the reference axes in inches as a real number (with decimal point) in columns 1-10. The plot is scaled so that half of this distance (from the centre of the plot to the end of the axis) corresponds to a relative frequency (per degree) of 1/120. The second and third parameter cards are identical and each contains up to 20 numbers in contiguous 4-column fields, either as right-justified integers or with decimal points. Each number is the parameter of the von Mises density function to be used as the weighting function. One rose diagram is produced for each field up to and including the last non-blank field. If a blank or zero field occurs before the last non-blank field, the data are plotted without smoothing for that field.

Output:

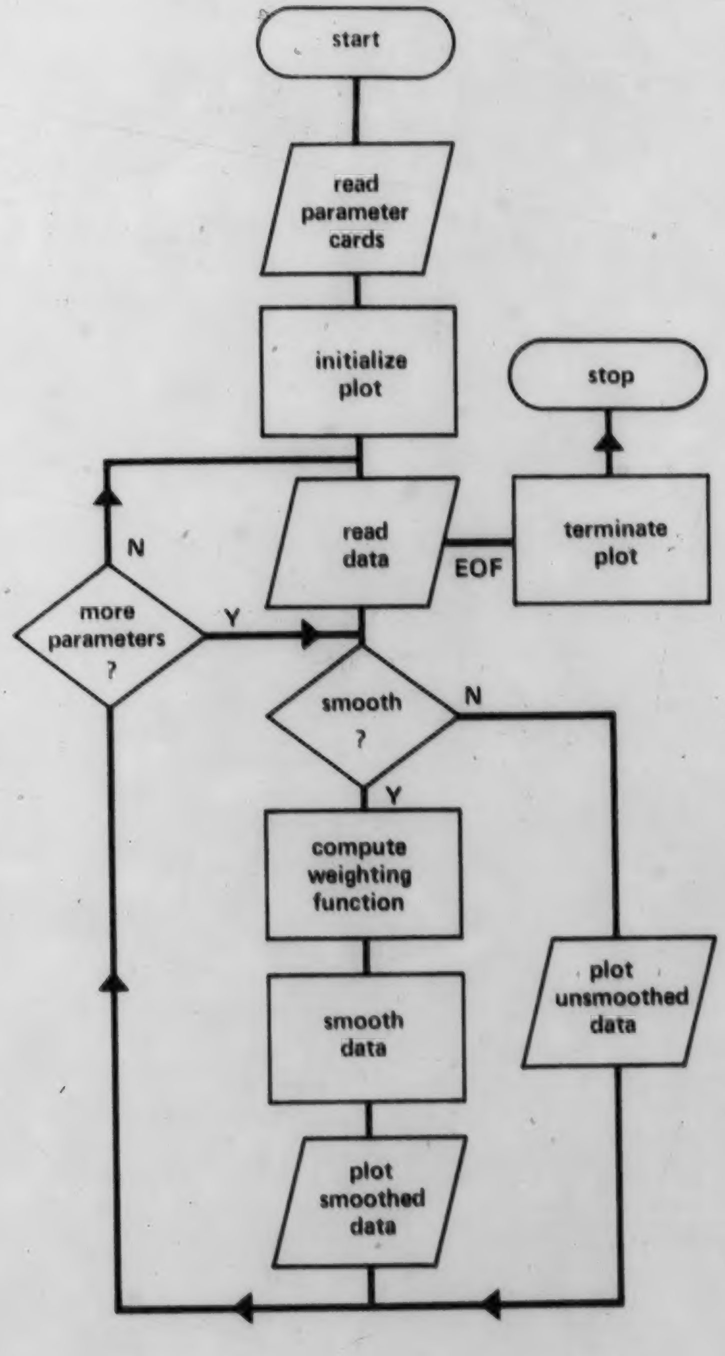
I/O unit 6: messages.

The title and number of observations for each set of data are written on this unit.

I/O unit ?: plot description file.

The number of this unit and the nature of the output depends on the system plot routines.

```
C PROGRAM ROSE - SMOOTHED CIRCULAR HISTOGRAMS.
      REAL T1(3)/' UNS','MOOT','HED '/,T2(3)/' SMO','OTHE','D K='/
     *,F(360),S(360),TITLE(30),A(20),B(20),BLANK/'    '/,W(90)
C
    1 FORMAT(20A4)
    2 FORMAT(1X,'*** ERROR - PARAMETEP CARD BLANK')
    3 FOPMAT(20F4.0)
    4 FORMAT(1X,I4,1X,30A4)
    5 FORMAT(F10.0)
C-----READ PARAMETER CARDS
      READ(5,5)DIA
      READ(5,1)A
      DO 100 J=1,20
      JJ=21-J
      IF(A(JJ).NE.BLANK)GOTO200
  100 CONTINUE
      WRITE(6,2)
      STOP
  200 CONTINUE
      READ(5,3)B
C-----INITIALIZE PLOT
      CALL ROSEC2(TITLE,N,F,1,DIA)
C-----READ DATA
  250 READ(1,END=990)TITLE,N,F
      WRITE(6,4)N,TITLE
      DO 600 J=1,JJ
      IF(B(J).NE.0.0)GOTO300
C-----PRODUCE UNSMOOTHED PLOT
      DO 260 I=1,3
  260 TITLE(I+11)=T1(I)
      TITLE(15)=BLANK
      CALL ROSEC2(TITLE,N,F,2,DIA)
      GOTO600
C-----PRODUCE SMOOTHED PLOT
  300 CALL WTFN(B(J),W,NW)
      CALL SMTH(F,S,W,NW)
      DO 350 I=1,3                                                      ROSE 370
  350 TITLE(I+11)=T2(I)                                                 ROSE 380
      TITLE(15)=A(J)                                                    ROSE 390
      CALL ROSEC2(TITLE,N,S,2,DIA)                                      ROSE 400
  600 CONTINUE                                                          ROSE 410
      GOTO250                                                           ROSE 420
C-----CLOSE PLOT                                                        ROSE 430
  990 CALL ROSEC2(TITLE,N,F,0,DIA)                                      ROSE 440
      STOP                                                              ROSE 450
C     DEBUG SUBCHK,TRACE                                                ROSE 460
C     AT 200                                                            ROSE 470
C     TRACE ON                                                          ROSE 480
      END                                                               ROSE 490
      SUBROUTINE ROSEC2(TITLE,N,F,J,DIA)                                ROSE 500
C OPENS (J=1) AND CLOSES (J=0) THE PLOT FILE AND DRAWS THE              ROSE 510
C HISTOGRAMS (J=2).                                                     ROSE 520
      REAL TITLE(15),F(360),A(360),DTR/1.745329E-2/                     ROSE 530
C                                                                       ROSE 540
      IF(J-1)900,100,200                                                ROSE 550
  100 CALL PLOTS                                                        ROSE 560
      CALL PLOT(-4.0,23.0,-3)                                           ROSE 570
      K=3                                                               ROSE 580
      DO 110 I=1,360                                                    ROSE 590
  110 A(I)=DTR*(90-I)                                                   ROSE 600
      RETURN                                                            ROSE 610
  200 K=K+1                                                             ROSE 620
      IF(K-3)210,210,220                                                ROSE 630
  210 CALL PLOT(0.0,9.0,-3)                                             ROSE 640
      GOTO300                                                           ROSE 650
  220 K=1                                                               ROSE 660
      CALL PLOT(9.0,-18.0,-3)                                           ROSE 670
  300 CONTINUE                                                          ROSE 680
      R=DIA/2.0                                                         ROSE 690
      H=R/20.0                                                          ROSE 700
      CALL SYMBOL(-R,-R-H-H,H,TITLE,0.0,60)                             ROSE 710
      AN=N                                                              ROSE 720
      FACT=1.0/(120.0*R)                                                ROSE 730
      CALL SYMBOL(-R,-R,H,'N=',0.0,2)                                   ROSE 740
      CALL NUMBER(-R+H+H,-R,H,AN,0.0,-1)                                ROSE 750
      CALL POLAR(F,A,360,1,-1,0.0,FACT)                                 ROSE 760
      CALL PLOT(0.0,R,3)                                                ROSE 770
      CALL PLOT(0.0,-R,2)                                               ROSE 780
      CALL PLOT(R,0.0,3)                                                ROSE 790
      CALL PLOT(-R,0.0,2)                                               ROSE 800
      RETURN                                                            ROSE 810
  900 CALL WHERE(X,Y,FA)                                                ROSE 820
      CALL PLOT(X,Y,999)                                                ROSE 830
      RETURN                                                            ROSE 840
      END                                                               ROSE 850
      SUBROUTINE WTFN(K,W,I)                                            ROSE 860
C COMPUTES THE WEIGHTING FUNCTION W GIVEN THE VON MISES                 ROSE 870
C PARAMETER K.                                                          ROSE 880
      REAL K,W(90)                                                      ROSE 890
      DO 50 I=1,90                                                      ROSE 900
   50 W(I)=0.0                                                          ROSE 910
      W(1)=1.0                                                          ROSE 920
      I=2                                                               ROSE 930
      A=1.745329E-2                                                     ROSE 940
  100 D=EXP(K*(COS(A)-1.0))                                             ROSE 950
      IF(D.LT.1.0E-6)GOTO300                                            ROSE 960
      W(I)=D                                                            ROSE 970
      I=I+1                                                             ROSE 980
      IF(I.GT.90)GOTO300                                                ROSE 990
      A=A+1.745329E-2                                                   ROSE1000
      GOTO100                                                           ROSE1010
  300 I=I-1                                                             ROSE1020
      RETURN                                                            ROSE1030
      END                                                               ROSE1040
      SUBROUTINE SMTH(F,S,W,NW)                                         ROSE1050
C COMPUTES THE SMOOTHED DENSITY FUNCTION S GIVEN THE UNSMOOTHED         ROSE1060
C DENSITY FUNCTION F AND THE WEIGHTING FUNCTION W.                      ROSE1070
      REAL F(360),S(360),W(90)                                          ROSE1080
      SR=0.0                                                            ROSE1090
      K=1                                                               ROSE1100
   80 C=F(K)*W(1)                                                       ROSE1110
      L=1                                                               ROSE1120
  100 K1=K-L                                                            ROSE1130
      IF(K1.LT.1)K1=K1+360                                              ROSE1140
      K2=K+L                                                            ROSE1150
      IF(K2.GT.360)K2=K2-360                                            ROSE1160
      C=C+W(L+1)*(F(K1)+F(K2))                                          ROSE1170
      L=L+1                                                             ROSE1180
      IF(L.EQ.NW)GOTO200                                                ROSE1190
      IF(W(L+1).NE.0.0)GOTO100                                          ROSE1200
  200 S(K)=C                                                            ROSE1210
      SR=SR+C                                                           ROSE1220
      K=K+1                                                             ROSE1230
      IF(K.LE.360)GOTO80                                                ROSE1240
      DO 250 I=1,360                                                    ROSE1250
  250 S(I)=S(I)/SR                                                      ROSE1260
      RETURN                                                            ROSE1270
      END                                                               ROSE1280
```


start
read parameter cards
initialize plot
stop
read data
EOF
terminate plot
N
more parameters ?
Y
smooth ?
N
Y
compute weighting function
smooth data
plot smoothed data
plot unsmoothed data

Flow Diagram for Program ROSE

Identification

Program name:

SAMPLEGEN

Program title:

Simulated samples from Fisher, great circle and uniform distributions.

Author:

John Ramsden

Organization:

Department of Geology, University of Alberta, Edmonton, Alberta, Canada

Date:

1 October 1974

Program version number:

1

Application

Purpose:

To generate simulated random samples from Fisher, great-circle and uniform distributions on the sphere.

Method:

For the Fisher and great-circle distributions, plunges are generated using algorithms given by Stauffer, MR, 1966, An empirical-statistical study of three-dimensional fabric diagrams as used in structural analysis, Can Jour Earth Sci, 3, 473-498. For the uniform distribution, plunges are generated so that their sines have a uniform distribution on the interval (-1,+1). In all cases, trends are generated from a uniform distribution on (0,2).

Data restrictions:

Maximum size of samples is 1000 points. For the Fisher distribution, the concentration parameter must not exceed 174.

Timing:

0.98 seconds of CPU time were required to generate 5 samples of 100 points each from a Fisher distribution with parameter of 100.

Implementation

Original installation:

Organization: University of Alberta Computing Services.
Computer: IBM 360 model 67 (1024K).
I/O devices: card, disc, tape, typewriter terminals and display terminals.
Software: Michigan Terminal System (MTS), a virtual-storage time-sharing system.

Source language:

FORTRAN IV

Subroutines required:

Requires system function subprogram URAND.

Size:

Size of user-supplied program = 14000 bytes. Size of system routine = 88 bytes. Total size = 14088 bytes.

Requirements for I/O devices:

I/O unit no.	read/ write	Maximum FORTRAN record length	device control statements	suggested device
1	write	12128 unformatted	none	tape, disc
5	read	16 formatted	none	card reader
6	write	49 formatted	none	line printer
7	write	7 formatted	none	disc

Usage

Input:

I/O unit 5: parameters.

One card containing

ccl	contents
1	1 for Fisher, 2 for great circle, 3 for uniform.
2-5	number of points in sample, right-justified integer.
6-9	distribution parameter, right-justified integer or real.
10-13	distribution parameter repeated, read as characters.
14-16	number of samples, right-justified integer.

Output:

I/O unit 1: generated samples.

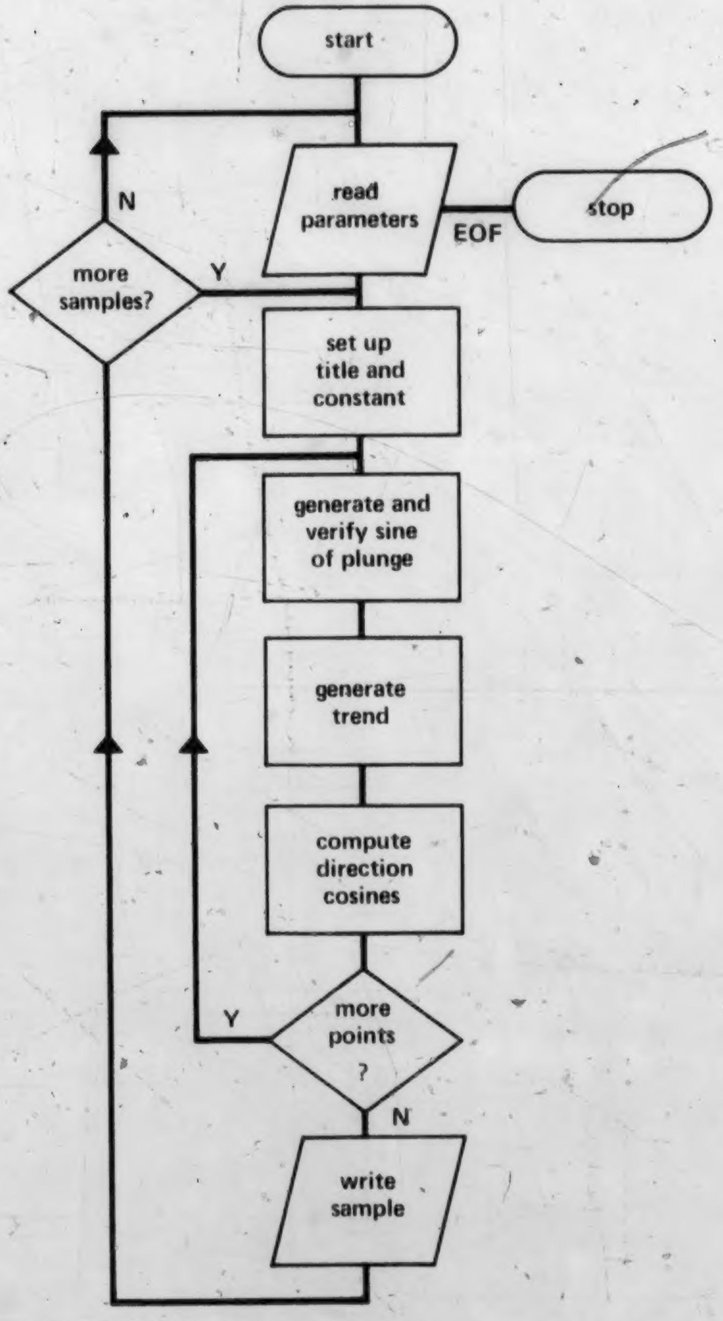
One unformatted record for each sample generated consisting of 4 bytes containing the number of points in the sample as a fullword integer, 120 bytes containing the title, 4 bytes containing the number of direction cosines in the record, and from 12 to 12000 bytes containing the direction cosines of the data points as one-word real numbers in the sequence a1, b1, g1, a2, b2, g2, . . .

I/O unit 6: messages.

I/O unit 7: heading.

One record containing the distribution parameter and sample size (for use by other programs).

```
C  PROGRAM NAME - SAMPLEGEN (SMPG)                                         SMPG  10
C PURPOSE:                                                                 SMPG  20
C      TO GENERATE RANDOM SAMPLES FROM FISHER, GIRDLE, OR                  SMPG  30
C      UNIFORM DISTRIBUTIONS ON THE SPHERE IN A FORM SUITABLE FOR          SMPG  40
C      IMMEDIATE PROCESSING BY OTHER PROGRAMS IN THE ODPRO PACKAGE.        SMPG  50
C SUBROUTINES REQUIRED:                                                    SMPG  60
C      FUNCTION SUBPROGRAM URAND FROM *LIBRARY                             SMPG  70
C I/O UNITS REFERENCED:                                                    SMPG  80
C      1: OUT: SIMULATED SAMPLE                                            SMPG  90
C      5: IN: PARAMETERS                                                   SMPG 100
C      7: OUT: HEADING                                                     SMPG 110
C                                                                          SMPG 120
      INTEGER ID1(30)/'SIMU','LATE','D RA','NDOM',' SAM','PLE ','FROM',    SMPG 130
     *' FIS','HER ','DIST','RIBU','TION','  K=',17*'    '/                 SMPG 140
      INTEGER ID2(30)/'SIMU','LATE','D RA','NDOM',' SAM','PLE ','FROM',    SMPG 150
     $' GIR','DLE ','DIST','RIBU','TION','  K=',17*'    '/                 SMPG 160
      INTEGER ID3(30)/'SIMU','LATE','D RA','NDOM',' SAM','PLE ','FROM',    SMPG 170
     *' UNI','FORM',' DIS','TRIB','UTIO','N   ',17*'    '/                 SMPG 180
      INTEGER Z/0/,CHAR                                                    SMPG 190
      REAL E/2.71828/,DC(3000)                                             SMPG 200
C READ PARAMETER CARD                                                      SMPG 210
   10 READ(5,30,END=700)KLUE,NUM,XK,CHAR,NSAM                              SMPG 220
      WRITE(7,1)XK,NUM                                                     SMPG 230
    1 FORMAT('K=',F5.1/'N=',I4)                                            SMPG 240
   30 FORMAT(I1,I4,F4.0,A4,I3)                                             SMPG 250
      DO 625 ISAM=1,NSAM                                                   SMPG 260
  100 GO TO (110,210,405),KLUE                                             SMPG 270
C SET UP TITLE AND CONSTANT                                                SMPG 280
  110 EK=E**XK                                                             SMPG 290
  120 ID1(14)=CHAR                                                         SMPG 300
  200 GO TO 405                                                            SMPG 310
  210 ID2(14)=CHAR                                                         SMPG 320
  405 K=0                                                                  SMPG 330
  410 DO 560 I=1,NUM                                                       SMPG 340
  415 K=K+3                                                                SMPG 350
  420 R=URAND(Z)                                                           SMPG 360
  430 GO TO (440,460,480),KLUE                                        SMPG 370
C GENERATE SIN OF PLUNGE FROM FISHER DISTRIBUTION                     SMPG 380
  440 SP=ALOG((1.-R)*EK+R/EK)/XK                                      SMPG 390
  450 GO TO 490                                                       SMPG 400
C GENERATE SIN OF PLUNGE FROM GIRDLE DISTRIBUTION                     SMPG 410
  460 SP=TAN((1.-2.*R)*ATAN(XK))/XK                                   SMPG 420
  470 GO TO 490                                                       SMPG 430
C GENERATE SIN OF PLUNGE FROM UNIFORM DISTRIBUTION                    SMPG 440
  480 SP=R*2.-1.                                                      SMPG 450
C VERIFY SIN OF PLUNGE                                                SMPG 460
  490 IF(SP.GE.1.)WRITE(6,491)I,SP                                    SMPG 470
  491 FORMAT(//1X,'*****WARNING*****'/                                SMPG 480
     *1X,'PROGRAM ERROR - SP>1 - I=', I4,5X, 'SP=',E11.4)             SMPG 490
C COMPUTE COSINE OF PLUNGE, TREND, AND DIRECTION COSINES.             SMPG 500
  500 CP=SQRT(1.-SP*SP)                                               SMPG 510
  510 R=URAND(Z)                                                      SMPG 520
  520 T=R*6.283185                                                    SMPG 530
  530 DC(K-2)=COS(T)*CP                                               SMPG 540
  540 DC(K-1)=SIN(T)*CP                                               SMPG 550
  550 DC(K)=SP                                                        SMPG 560
  560 CONTINUE                                                        SMPG 570
C WRITE SAMPLE                                                        SMPG 580
  570 GO TO (580,600,620),KLUE                                        SMPG 590
  580 WRITE(1)NUM,ID1,K,(DC(I),I=1,K)                                 SMPG 600
  590 GO TO 625                                                       SMPG 610
  600 WRITE (1)NUM,ID2,K,(DC(I),I=1,K)                                SMPG 620
  610 GO TO 625                                                       SMPG 630
  620 WRITE(1)NUM,ID3,K,(DC(I),I=1,K)                                 SMPG 640
  625 CONTINUE                                                        SMPG 650
  630 GO TO 10                                                        SMPG 660
  700 STOP                                                            SMPG 670
      END                                                             SMPG 680
```


start
read parameters
EOF
stop
N
more samples?
Y
set up title and constant
generate and verify sine of plunge
generate trend
compute direction cosines
Y
more points ?
N
write sample

Flow Diagram for Program SAMPLEGEN

Identification

Program name:

TRVLST2

Program title:

Areal-analysis package - traverse list for subset of data file.

Author:

John Ramsden

Organization:

Department of Geology, University of Alberta, Edmonton, Alberta, Canada

Date:

1 October 1974

Program version number:

2

Application

Purpose:

To generate a traverse file for a subset of a data file.

Method:

The program reads traverse data from a traverse file. By simultaneously reading the data file, the program obtains the range of line numbers in the data file that correspond to observations on each traverse. A new traverse file is produced containing this information.

Data restrictions:

None.

Timing:

The program required 1.3 seconds of CPU time to process a data file containing 297 observations on 43 traverses. The program took 1.4 seconds of CPU time to process a data file containing 297 observations on 43 traverses.

Implementation

Original installation:

Organization: University of Alberta Computing Services.
Computer: IBM 360 model 67 (1024K).
I/O devices: card, disc, tape, typewriter terminals and display terminals.
Software: Michigan Terminal System (MTS), a virtual-storage time-sharing system.

Source language:

FORTRAN IV

Subroutines required:

None.

Size:

2464 bytes. 2464 bytes.

Requirements for I/O devices:

I/O unit no.	read/ write	Maximum FORTRAN record length (bytes)	device control statements	suggested device
1	read	808 unformatted	none	tape; disc
4	read	determined by format	none	tape; disc; card reader.
8	write	determined by format	none	tape; disc; card punch.

Usage

Input:

I/O unit 1: traverse identification from data file.

As written by program INPUT1 or INPUTK1.

I/O unit 4: traverse list.

Identical records, one for each traverse. Variables read are: 1 double-word variable which must contain a 1- to 8-character identification for the traverse, followed by 11 single-word variables which may contain any other information about the traverse.

I/O unit 5: formats for reading and writing the traverse files.

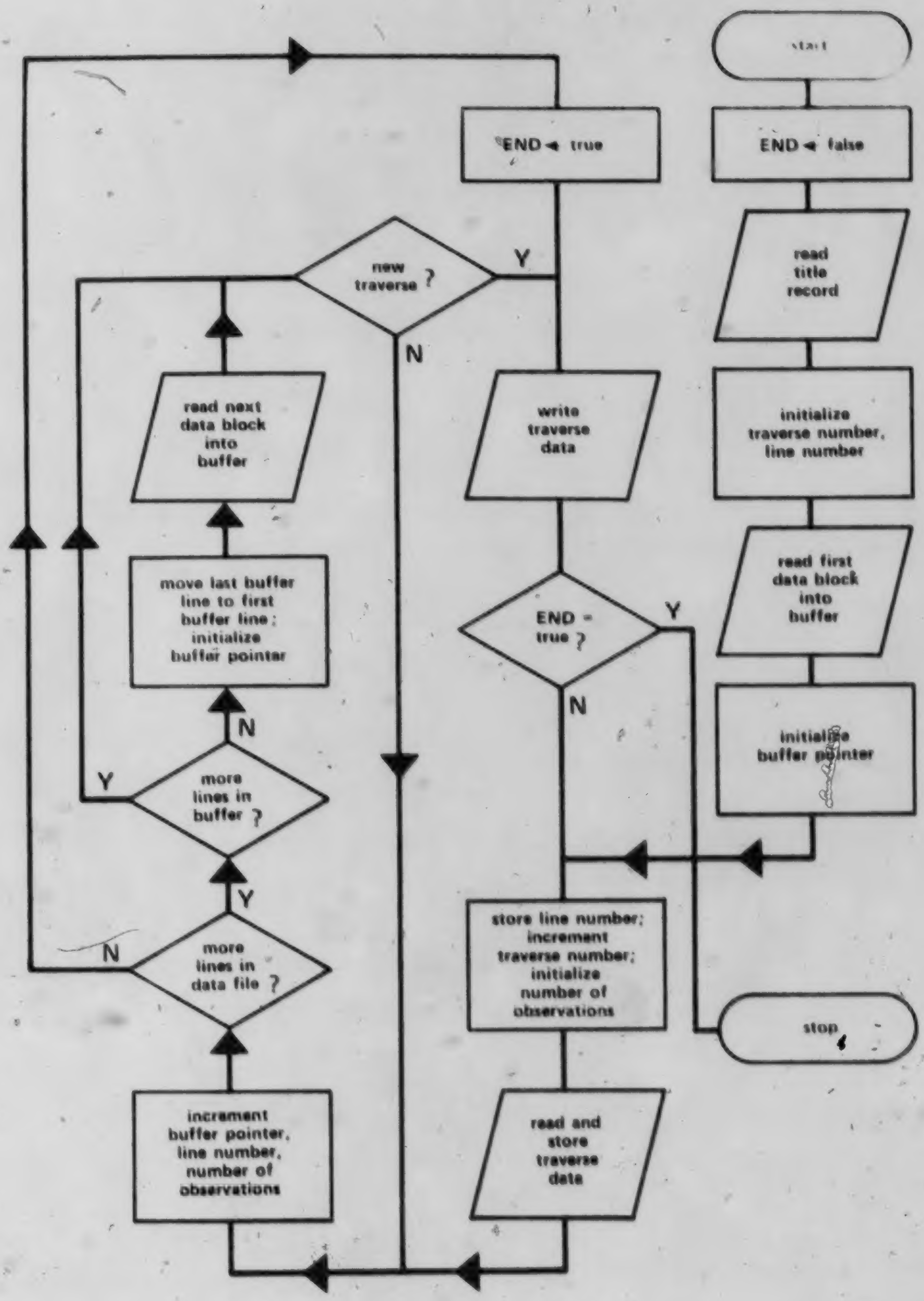
Two records, one format in each record, each format in columns 3 to 102.

Output:

I/O unit 8: new traverse file.

Same variables as read on unit 4 preceded by 4 fullword integer variables containing (1) sequence number of traverse in input traverse file, (2) data-file line number of first observation on this traverse, (3) data-file line number of last observation on this traverse, (4) number of observations on this traverse.

```
C
C
C AREAL-ANALYSIS PACKAGE - PROGRAM "TRVLST2".
C
C
      REAL*8 ID,BUF(101),TRDATA(11)
      REAL BUF2(200)
      LOGICAL*1 END
      INTEGER*2 LINTYP,DEE/'D'/
      INTEGER TITLE(30),BL/'    '/,FMT(25),FMT2(25)
      EQUIVALENCE (BUF2,BUF(2))
      INTEGER APP(6)/'  TR','AVER','SES ','    ','    ','    '/
C
    2 FORMAT('T',30A4)
    3 FORMAT(A1,A3,I4)
    7 FORMAT(2X,25A4)
    5 FORMAT(I1)
  601 FORMAT(1X,A8)
C
      READ(5,7)FMT,FMT2
C
  530 END=.FALSE.
C
C-----READ & WRITE TITLE
C
      READ(1)NN,NV,NO,NPHR,TITLE
      DO 20 I=1,6
   20 TITLE(I+10)=APP(I)
      WRITE(8,2)TITLE
      WRITE(6,2)TITLE
C
C-----INITIALIZE TRAVERSE NUMBER, LINENUMBER AND BUFFER POINTER
C
      ITRAV=0
      LNR=1
      IBUF=2
C
C-----READ FIRST DATA BLOCK INTO BUFFER
C
      READ(1)N,IE,(BUF2(I),I=1,IE)
C
C-----STORE LINENUMBER AND INITIALIZE
C-----NUMBER OF OBSERVATIONS
C
   60 L1=LNR
      NOBS=1
C
C-----GET TRAV INFO FROM TRAVERSE FILE
C
   80 READ(4,FMT,END=450)ID,TRDATA
      ITRAV=ITRAV+1
      IF(ID.NE.BUF(IBUF))GOTO80
C
C-----INCREMENT COUNTERS
C
  100 IBUF=IBUF+1
      LNR=LNR+1
      NOBS=NOBS+1
      IF(LNR.GT.NO)GOTO500
      IF(IBUF.LE.101)GOTO400
      BUF(1)=BUF(101)
      IBUF=2
      READ(1)N,IE,(BUF2(I),I=1,IE)
  400 IF(BUF(IBUF).EQ.BUF(IBUF-1))GOTO100
      GOTO600
  500 END=.TRUE.
C
C-----WRITE TRAVERSE DATA
C
  600 CONTINUE
  300 WRITE(8,FMT2)ITRAV,L1,LNR,NOBS,ID,TRDATA
      IF(END)STOP99
      GOTO60
C
  420 STOP
  430 STOP 12
  440 STOP 11
  450 STOP 10
      END
```


start
END ◄ false
read title record
initialize traverse number, line number
read first data block into buffer
initialize buffer pointer
store line number; increment traverse number; initialize number of observations
read and store traverse data
increment buffer pointer, line number, number of observations
more lines in data file ?
Y
N
more lines in buffer ?
Y
N
move last buffer line to first buffer line; initialize buffer pointer
read next data block into buffer
new traverse ?
Y
N
END ◄ true
write traverse data
END = true ?
Y
N
stop

FLOW DIAGRAM FOR PROGRAM TRVLST2

Identification

Program name:

WNDPLT3

Program title:

Areal-analysis package - line-printer orientation diagram.

Author:

John Ramsden

Organization:

Department of Geology, University of Alberta, Edmonton, Alberta, Canada

Date:

1 October 1974

Program version number:

3

Application

Purpose:

For weighted, non-directed orientations, to produce on the line-printer equal-area projections of point density having a high density of counting locations.

Method:

The plane of the projection is horizontal, unless a parameter card is given redefining the projection plane. In this case the directions are rotated before being plotted. The program uses a 1 per cent counting circle, a contour interval of 1 per cent, and the following characters to represent successive contour intervals 0-1 per cent, 1-2 per cent, and so on " 1234567890abcdefghijklmnopqrstuvwxyz****". Any of these defaults may be overridden by appropriate parameter cards.

The input data must include weights. Unless otherwise directed, the program uses these weights in the computations. A parameter card may be used to specify that the weights are to be truncated at some specified upper limit, or that the weights are to be set equal to 1.

Data restrictions:

None.

Timing:

t=4+1.2s+0.006p; t=CPU time in seconds; s=number of samples (number of plots produced); p=total number of data points in all samples.

Implementation

Original installation:

Organization:	University of Alberta Computing Services.
Computer:	IBM 360 model 67 (1024K).
I/O devices:	card, disc, tape, typewriter terminals and display terminals.
Software:	Michigan Terminal System (MTS), a virtual-storage time-sharing system.

Source language:

FORTRAN IV

Subroutines and function subprograms required:

The following subroutines from JR subroutine library: DCAN ROTMAT VRM1. The following subroutines from SSP subroutine library: GMPRD GTPRD.

Size:

Total size of program including all required subroutines is 52752 bytes.

Requirements for I/O devices:

I/O unit no.	read/ write	Maximum FORTRAN record length (bytes)	device control statements	suggested device
1	read	1608 unformatted	none	tape; disc.
3	read	80 formatted	none	disc; card.
4	read	determined by given format	none	disc; card.
5	read	80 formatted	none	disc; card.
6	write	42 formatted	none	line printer
8	write	84 formatted	none	line printer
9	write	127 formatted	none	line printer

Usage

Input:

I/O unit 1: data

As written by program INPUT, INPUTK, or DSPDATA. As many samples as desired. One plot will be produced for each sample.

I/O unit 3: direction cosines of counting locations

These are contained in a deck of 314 cards in EBCDIC.

I/O unit 4: normal to projection plane

The contents and format of the single record that may be read from this unit are determined by the "p" parameter card.

I/O unit 5: parameters

All parameter cards are optional, all parameters having default values. The reading of parameter cards must be terminated by a blank record or an end-of-file. One parameter record must be supplied for each parameter for which a value is to be specified. The parameters with their defaults are:

(1) Normal to projection plane. Default: vertical. Card format:

col	contents
1	"p"
2-11	trend or first direction cosine
12-21	plunge or second direction cosine

22-31	blank or third direction cosine
32-40	code: +-1.0 if trend and plunge given in degrees +-2.0 if trend and plunge given in radians +-3.0 if direction cosines given code>0 if information is on parameter card code<0 if information to be read on unit 4
41-80	format for reading unit 4 if required

(2) Size of counting circle. Default: 1%. Card format:

col	contents
1	"S"
2-11	per cent size of circle

(3) Contour interval. Default: 1%. Card format:

col	contents
1	"I"
2-11	contour interval

(4) Plot characters.
Default: " 123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ*****".
Card format:

col	contents
1	"C"
41-80	plot characters

(5) Weighting procedure. Default: weights used as read. Card format:

col	contents
1	"W"
2-11	value

Action:

If value=0 all weights will be set to 1.0 before data is processed.

If value>0 weights greater than "value" will be set equal to "value". Weights less than or equal to "value" will not be changed.

If value<0 the default action will remain if effect.

(6) Printout. Default: overprinting and list of character values. Card format:

col	contents
1	"C"

Action: Overprinting and list of character values suppressed.

Output:

I/O unit 6: messages

Error messages are self-explanatory.

I/O unit 8: plot

Preceding the plot, 5 records containing: (1) the first 80 characters of the title, (2) the remainder of the title, (3) number of data points and total weight, (4) size of counting circle, and (5) contour interval and characters. The plot consists of 49 lines each containing 82 characters, the first character in each line being a carriage-control character.

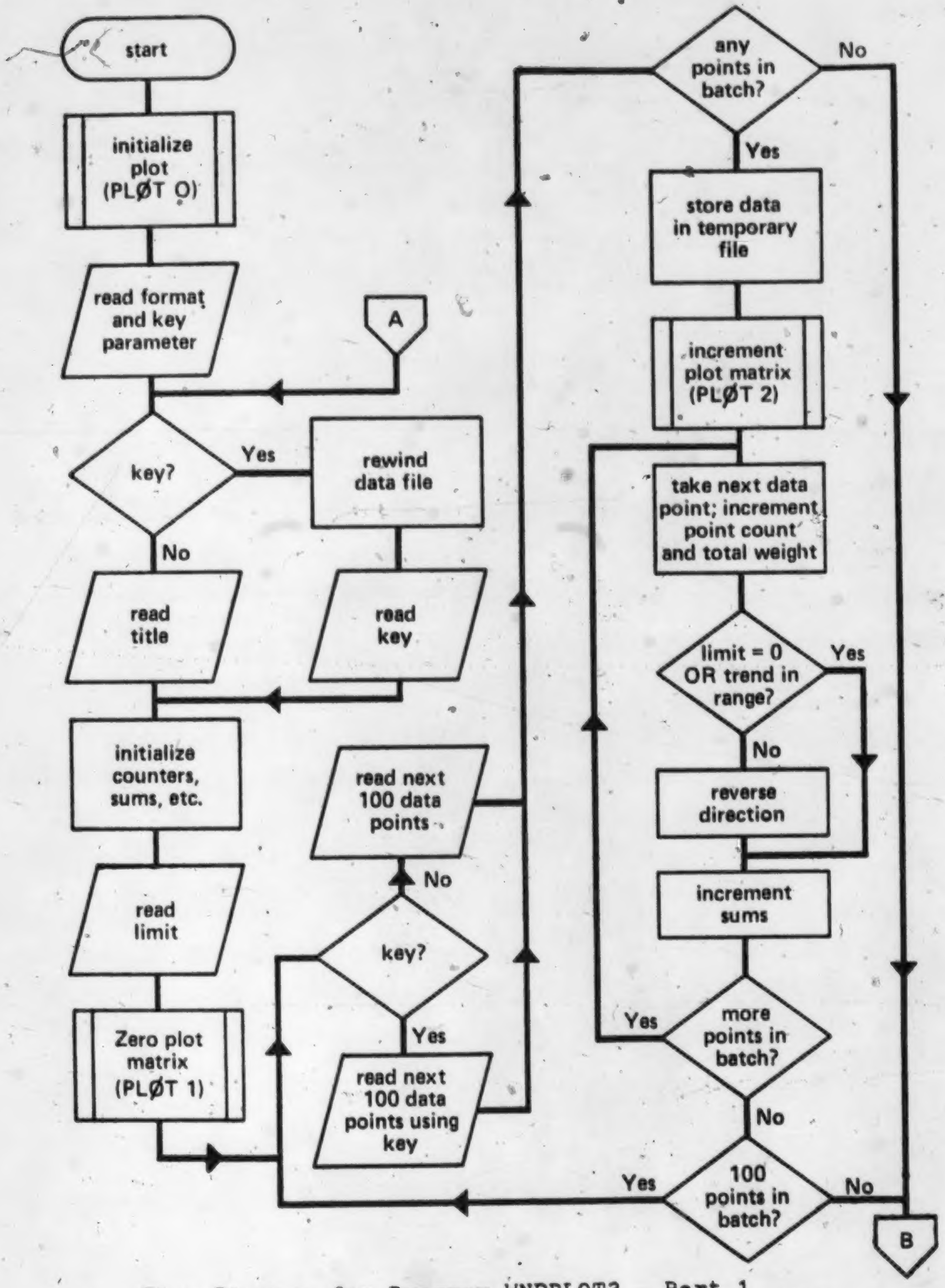
I/O unit 9: title of each sample

On this unit the program writes the number of data points and the title for each sample as it is read.

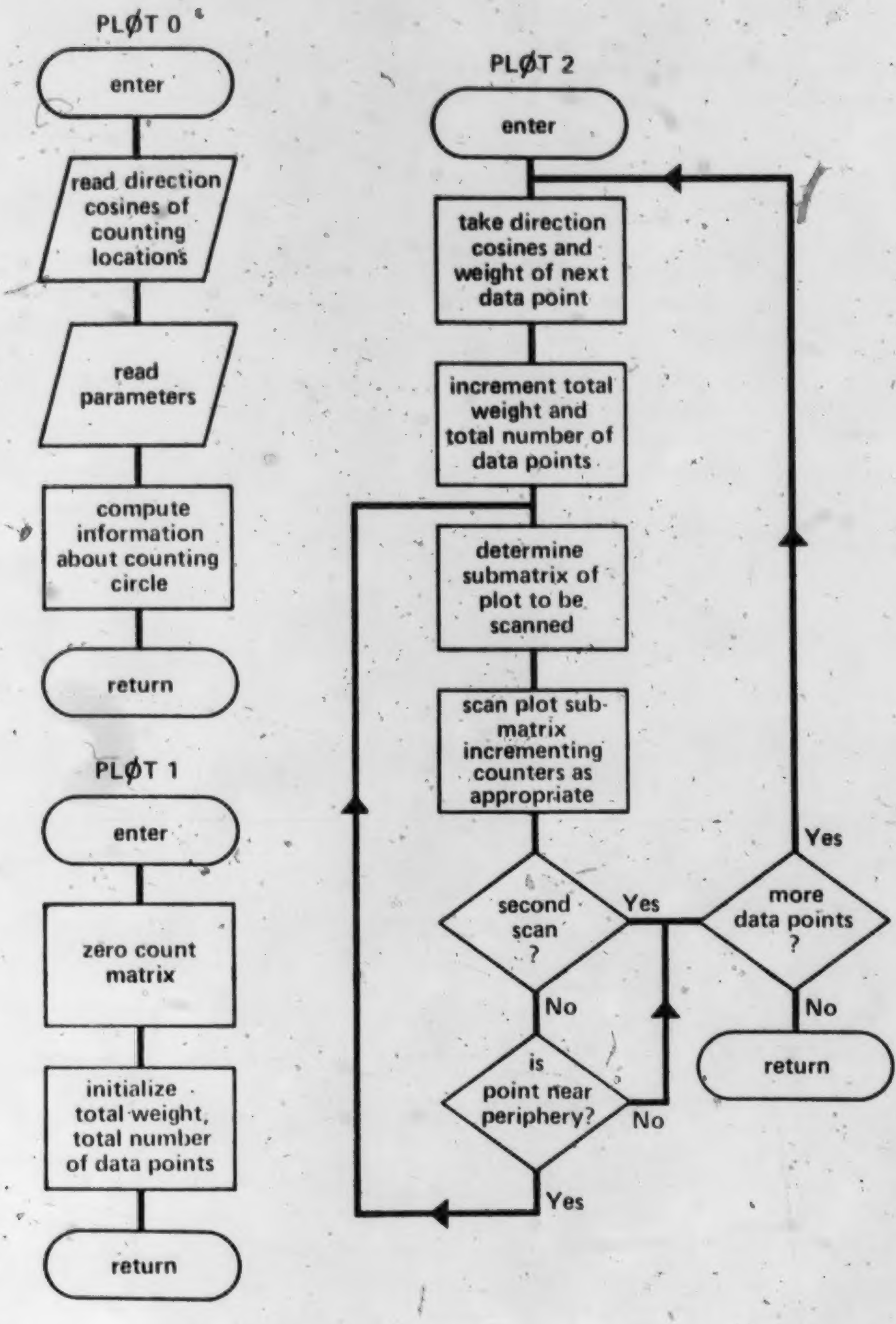
```
C
C
C AREAL-ANALYSIS PACKAGE - PROGRAM "WNDPLT3".
C
C
      REAL CONTI(10),FMT(15),COUNT(3713),MPLRC,DC(3,100),W(100),DCR(3,
     *100)
      INTEGER*2 ALOC(3713),BLOC(3713),GLOC(3713)
      INTEGER TITLE(30)
      LOGICAL*1 CHAR(40),CHAR1(40),ALPHA(40)
      REAL*8 CHA(5)/' 1234567','89ABCDEF','GHIJKLMN','OPQRSTUV',
     *'WXYZ****'/
      EQUIVALENCE (CHAR1,CHA)
      LOGICAL*1 ROTAT/.FALSE./,BOTH/.FALSE./
      REAL  PEE/'P'/,ESS/'S'/,EYE/'I'/,SEA/'C'/,BLANK/' '/,KEW/'Q'/
     *,X(4),RM(9),DUB/'W'/,OWE/'O'/
      COMMON/ONE/FMT,CONTI,COUNT,CHAR,ICODE1
      COMMON/TWO/SWT,NT,SIZE,RC,CRC,CRCC,MPLRC,ALOC,BLOC,GLOC
C
C
    1 FORMAT(A1,3F10.0,F9.0,40A1)
    2 FORMAT(/1X,'INVALID PARAMETER CARD'/1X,A1,4F10.4/
     *1X,'"',A1,'" IS INVALID')
    3 FORMAT(/1X,'INVALID PARAMETER CARD'/1X,A1,4F10.4/
     *1X,'"',F10.4,'" IS INVALID')
    4 FORMAT(40A1)
    5 FORMAT(1X,I5,4F10.4)
    6 FORMAT(I1)
    7 FORMAT(1X,I5,1X,30A4)
C
C
      ICODE1=1
      DO 8 I=1,40
    8 CHAR(I)=CHAR1(I)
      SIZE=1.0
      CONTI(1)=1.0
      WEIGHT=-1.0
   10 READ(5,1,END=90)A,X,ALPHA
      IF(A.EQ.PEE)GOTO19
      IF(A.EQ.KEW)GOTO20
      IF(A.EQ.DUB)GOTO35
      IF(A.EQ.ESS)GOTO40
      IF(A.EQ.EYE)GOTO60
      IF(A.EQ.SEA)GOTO80
      IF(A.EQ.BLANK)GOTO90
      IF(A.EQ.OWE)GOTO70
      WRITE(6,2)A,X,A
      STOP12
C
   19 BOTH=.TRUE.
      GOTO22
   20 ROTAT=.TRUE.
   22 J=X(4)
      IF(J.NE.0 .AND. IABS(J).LT.4)GOTO25
      WRITE(6,3)A,X,X(4)
      STOP12
   25 IF(J.GT.0)GOTO30
      J=-J
      GOTO(26,26,27),J
   26 READ(4,ALPHA)X(1),X(2)
      GOTO30
   27 READ(4,ALPHA)X(1),X(2),X(3)
      GOTO30
   30 CALL VPM1(X,RM,J)
      GOTO10
   35 WEIGHT=X(1)
      GOTO10
   40 SIZE=X(1)
      GOTO 10
   60 CONTI(1)=X(1)
      GOTO10
   70 ICODE1=0
      GOTO10
   80 DO 85 I=1,40
   85 CHAR(I)=ALPHA(I)
      GOTO10
C
C-----COMPUTE INFORMATION ABOUT COUNTING CIRCLE
C
   90 CONTINUE
      CALL GRID10(3)
      CRC=1.0-SIZE/100.0
      CRCC=CRC*10000.0
      RC=ARCOS(CRC)
      MPLRC=RC*3.937
C
C-----READ NEXT SAMPLE
C
  100 READ(1,END=300)NN,NV,NO,NPHR,TITLE
      WRITE(9,7)NO,TITLE
  110 CALL PLOT1A
      DO 150 LOOP=1,NPHR
      READ(1)NB,NWD,((DC(I,J),I=1,3),W(J),J=1,NB)
      DO 109 J=1,NB
      IF(WEIGHT)109,108,107
  107 IF(W(J).GT.WEIGHT)W(J)=WEIGHT
      GOTO109
  108 W(J)=1.0
  109 CONTINUE
      IF(.NOT.ROTAT)GOTO125
      CALL GMPRD(RM,DC,DCR,3,3,NB)
      CALL PLOT1B(DCR,W,NB)
      GOTO127
  125 CALL PLOT1B(DC,W,NB)
  127 CONTINUE
  150 CONTINUE
C
C-----PRODUCE PLOT
C
  200 CONTINUE
      CALL PLOT1C(8,TITLE)
      IF(.NOT.BOTH)GOTO100
      IF(ROTAT)GOTO250
      ROTAT=.TRUE.
      GOTO110
  250 ROTAT=.FALSE.
      GOTO100
C
  300 STOP
```

```
      END
      SUBROUTINE PLOT1A
C SETS COUNTERS TO ZERO
      REAL MPLRC,SUMW,COUNT(3713),CONTI(10),FMT(15)
      LOGICAL*1 CHAR(4,10)
      COMMON /ONE/ FMT,CONTI,COUNT,CHAR,ICODE1
      INTEGER*2 ALOC(3713),BLOC(3713),GLOC(3713)
      COMMON /TWO/ SUMW,NTOT,PCA,RC,CRC,CRCC,MPLRC,ALOC,BLOC,GLOC
C                     FILL 'COUNT' WITH ZEROS
    7 DO 10 I=1,3713
   10 COUNT(I)=0.0
C
      SUMW=0.0
      NTOT=0
      RETURN
C     DEBUG SUBTRACE,SUBCHK
      END
      SUBROUTINE PLOT1B(DC,W,N)
C INCREMENTS COUNTERS FOR A BATCH OF UP TO 100 DATA POINTS
      REAL DC(3,100),W(100)
      REAL MPLRC,MPLRC2,COUNT(79,47),FMT(15),CONTI(10),CHAR(10)
      INTEGER*2 ALOC(79,47),BLOC(79,47),GLOC(79,47),GLOCIJ
      COMMON /ONE/ FMT,CONTI,COUNT,CHAR,ICODE1
      COMMON /TWO/ SUMW,NTOT,PCA,RC,CRC,CRCC,MPLRC,ALOC,BLOC,GLOC
C
      DO 200 JJ=1,N
      WW=W(JJ)
      SUMW=SUMW+WW
      NTOT=NTOT+1
      A=DC(1,JJ)
      B=DC(2,JJ)
      G=DC(3,JJ)
      IF(G.GT.0.999999)G=0.999999
      IF(G.LT.-0.999999)G=-0.999999
      CP=SQRT(1.0-G*G)
      IF(CP.EQ.0.0)GO TO 100
      CT=A/CP
      ST=B/CP
      GO TO 110
  100 ST=0.0
      CT=1.0
  110 CONTINUE
      CALL RANGEX(ST,CT,G,MPLRC,I1,I2,J1,J2)
C     WRITE(6,1)ST,CT,G,MPLRC,I1,I2,J1,J2
C   1 FORMAT(1X,4F10.4,4I5)
   20 DO 40 J=J1,J2
      DO 40 I=I1,I2
      GLOCIJ=GLOC(I,J)
      IF(GLOCIJ.GT.10000)GO TO 40
      CPHI=ABS(A*ALOC(I,J)+B*BLOC(I,J)+G*GLOCIJ)
      IF(CPHI.GE.CRCC)COUNT(I,J)=COUNT(I,J)+WW
   40 CONTINUE
      IF(CP.LE.CRC)GO TO 200
      IF(CP.EQ.1.0)GO TO 50
      ARG=CRC/CP
      IF(ARG.LE.1.0)GO TO 45
C     WRITE(6,42)CRC,CP
C  42 FORMAT(1X,2F10.6)
      GO TO 200
   45 MPLRC2=ARCOS(ARG)*3.3937
```

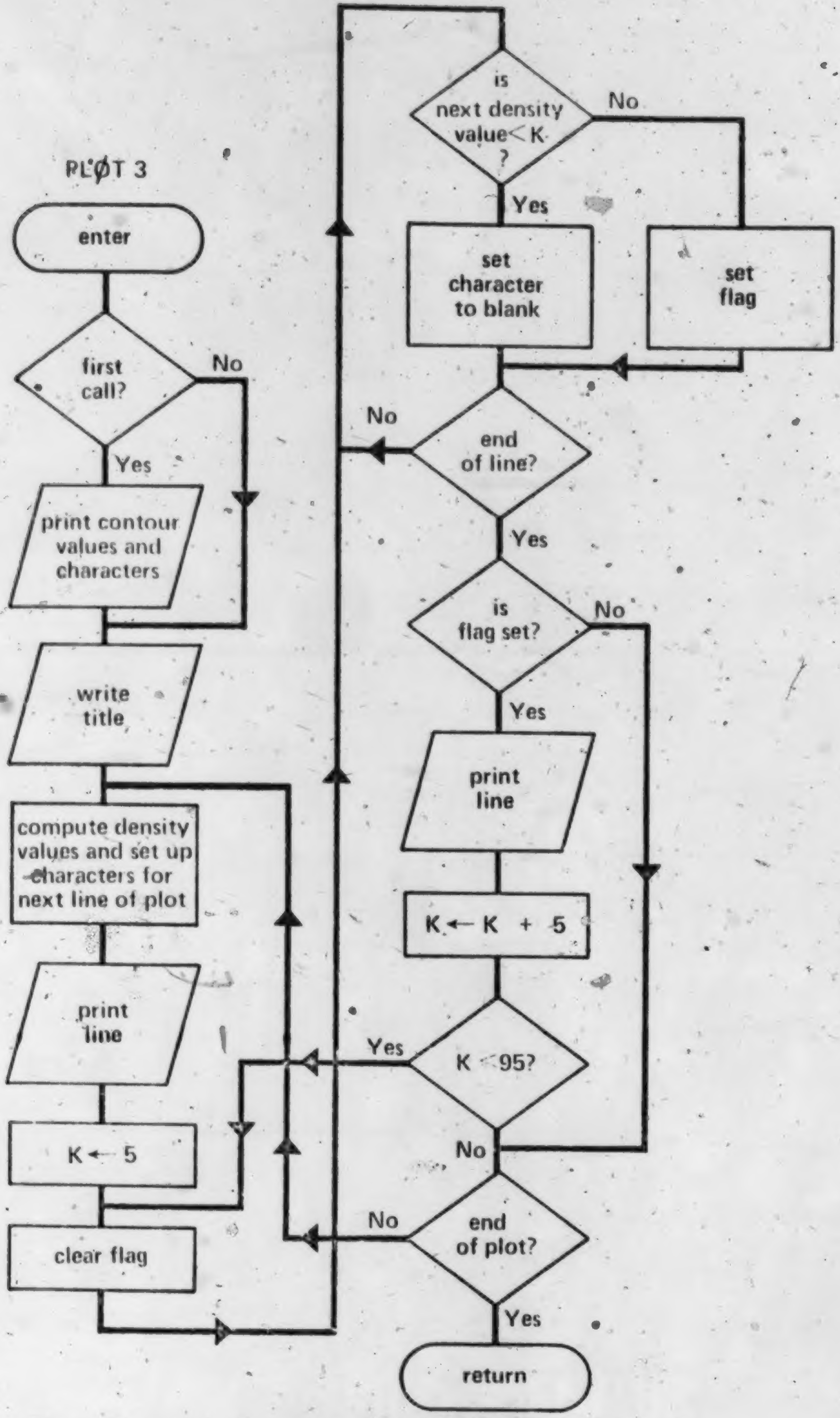
```
      GO TO 60                                                          WNDP1800
   50 MPLRC2=MPLRC                                                      WNDP1810
   60 ST=-ST                                                            WNDP1820
      CT=-CT                                                            WNDP1830
      CP=-10.0                                                          WNDP1840
      CALL RANGEX(ST,CT,0.0,MPLRC2,I1,I2,J1,J2)                         WNDP1850
      GO TO 20                                                          WNDP1860
  200 CONTINUE                                                          WNDP1870
      RETURN                                                            WNDP1880
      END                                                               WNDP1890
      SUBROUTINE PLOT1C(IOU,ID)                                         WNDP1900
C GENERATES PRINTER PLOT                                                WNDP1910
      INTEGER*2 ALOC(3713),BLOC(3713),GLOC(3713)                        WNDP1920
      REAL CONTI(10),COUNT(79,47),FMT(15),ID(30)                        WNDP1930
      LOGICAL*1 PLOT(79),CHAR(40),CENT/'+'/,BLANK/' '/                  WNDP1940
     *,CH(100)/100*'*'/                                                 WNDP1950
      INTEGER*2 NUM(79)                                                 WNDP1960
      COMMON/ONE/FMT,CONTI,COUNT,CHAR,ICODE1                            WNDP1970
      COMMON/TWO/SUMW,NTOT,PCA,RC,CRC,CRCC,MPLRC,ALOC,BLOC,GLOC         WNDP1980
      EQUIVALENCE (CI,CONTI)                                            WNDP1990
      LOGICAL DONT/.FALSE./                                             WNDP2000
C                                                                       WNDP2010
      IF(DONT.OR.ICODE1.EQ.0)GOTO105                                    WNDP2020
      WRITE(IOU,5)                                                      WNDP2030
      CONT=0.0                                                          WNDP2040
      DO 103 I=1,40                                                     WNDP2050
      IF(CONT.GT.100.0)GOTO104                                          WNDP2060
      WRITE(IOU,6)CHAR(I),CONT                                          WNDP2070
      CONT=CONT+CONTI(1)                                                WNDP2080
  103 CONTINUE                                                          WNDP2090
  104 DONT=.TRUE.                                                       WNDP2100
  105 CONTINUE                                                          WNDP2110
C                                                                       WNDP2120
      DO 10 I=1,40                                                      WNDP2130
   10 CH(I)=CHAR(I)                                                     WNDP2140
      FACT=100.0/(SUMW*CI)                                              WNDP2150
C                                                                       WNDP2160
      WRITE(IOU,4)ID,NTOT,SUMW,PCA,CI,CHAR                              WNDP2170
      WRITE(IOU,1)                                                      WNDP2180
C                                                                       WNDP2190
      JSTART=1                                                          WNDP2200
      JSTOP=23                                                          WNDP2210
  200 DO 280 J=JSTART,JSTOP                                             WNDP2220
      DO 220 I=1,79                                                     WNDP2230
      L=FACT*COUNT(I,J)+1.0                                             WNDP2240
      NUM(I)=L                                                          WNDP2250
      PLOT(I)=CH(L)                                                     WNDP2260
  220 CONTINUE                                                          WNDP2270
      WRITE(IOU,2)PLOT                                                  WNDP2280
      IF(ICODE1.EQ.0)GOTO280                                            WNDP2290
      DO 270 K=5,95,5                                                   WNDP2300
      CALL BLNK(NUM,PLOT,K,&280)                                        WNDP2310
      WRITE(IOU,12)PLOT                                                 WNDP2320
  270 CONTINUE                                                          WNDP2330
  280 CONTINUE                                                          WNDP2340
C                                                                       WNDP2350
      IF(JSTOP.EQ.47)GOTO400                                            WNDP2360
C                                                                       WNDP2370
      DO 320 I=1,79                                                     WNDP2380
      L=FACT*COUNT(I,24)+1.0                                            WNDP2390
      NUM(I)=L                                                          WNDP2400
      PLOT(I)=CH(L)                                                     WNDP2410
  320 CONTINUE                                                          WNDP2420
      PLOT(40)=CENT                                                     WNDP2430
      WRITE(IOU,3)PLOT                                                  WNDP2440
      IF(ICODE1.EQ.0)GOTO380                                            WNDP2450
      DO 370 K=5,95,5                                                   WNDP2460
      CALL BLNK(NUM,PLOT,K,&380)                                        WNDP2470
      WRITE(IOU,12)PLOT                                                 WNDP2480
  370 CONTINUE                                                          WNDP2490
  380 CONTINUE                                                          WNDP2500
C                                                                       WNDP2510
      JSTART=25                                                         WNDP2520
      JSTOP=47                                                          WNDP2530
      GOTO200                                                           WNDP2540
C                                                                       WNDP2550
  400 CONTINUE                                                          WNDP2560
      WRITE(IOU,1)                                                      WNDP2570
      RETURN                                                            WNDP2580
C                                                                       WNDP2590
    1 FORMAT(1X,1X,39X,'I')                                             WNDP2600
    2 FORMAT(1X,1X,79A1)                                                WNDP2610
    3 FORMAT(1X,'-',79A1,'-')                                           WNDP2620
    4 FORMAT('1',20A4/1X,10A4/1X,I4,' OBSERVATIONS WITH TOTAL WEIGHT OF'WNDP2630
     *,F8.1/1X,'PERCENT OF TOTAL WEIGHT IN',F5.1,' PERCENT OF AREA'/    WNDP2640
     *1X,'CONTOUR INTERVAL',F5.1,'   CHARACTER SEQUENCE ',40A1)         WNDP2650
    5 FORMAT('1'/' '/' '/' CONTOUR VALUES AND CHARACTERS FOR THIS',     WNDP2660
     *' RUN ARE'/' '/' '/' ')                                           WNDP2670
    6 FORMAT(5X,A1,' > ',F5.1,'%')                                      WNDP2680
   12 FORMAT('+',1X,79A1)                                               WNDP2690
      END                                                               WNDP2700
      SUBROUTINE GRID10(I1)                                             WNDP2710
C READS DIRECTION COSINES OF COUNTING LOCATIONS                         WNDP2720
      INTEGER*2 ALOC(3713),BLOC(3713),GLOC(3713),KA,KB                  WNDP2730
      COMMON /TWO/ SUMW,NTOT,PCA,RC,CRC,CRCC,MPLRC,ALOC,BLOC,GLOC       WNDP2740
      READ(I1,6)                                                        WNDP2750
    6 FORMAT(///)                                                       WNDP2760
      DO 5 J=1,310                                                      WNDP2770
      READ(I1,500)KA,KB,(ALOC(I),BLOC(I),GLOC(I),I=KA,KB)               WNDP2780
  500 FORMAT(38A2)                                                      WNDP2790
    5 CONTINUE                                                          WNDP2800
      RETURN                                                            WNDP2810
C     DEBUG SUBTRACE,SUBCHK                                             WNDP2820
      END                                                               WNDP2830
      SUBROUTINE RANGEX(ST,CT,G,R,I1,I2,J1,J2)                          WNDP2840
C COMPUTES THE SUBSET OF THE COUNTING LOCATION ARRAY TO BE SCANNED      WNDP2850
      U=3.937                                                           WNDP2860
      IF(G.NE.0.0)U=3.937*SQRT(1.0-G)                                   WNDP2870
      X=4.0+U*ST                                                        WNDP2880
      Y=4.0-U*CT                                                        WNDP2890
      X1=X-R                                                            WNDP2900
      IF(X1.LT.0.0)X1=0.0                                               WNDP2910
      I1=X1*10.0+1.0                                                    WNDP2920
      X2=X+R                                                            WNDP2930
      IF(X2.GE.8.0)X2=7.95                                              WNDP2940
      I2=X2*10.0                                                        WNDP2950
      Y1=Y-R                                                            WNDP2960
      IF(Y1.LT.0.0)Y1=0.0                                               WNDP2970
      J1=Y1*6.0+1.0                                                     WNDP2980
      Y2=Y+R                                                            WNDP2990
      IF(Y2.GE.8.0)Y2=7.9                                               WNDP3000
      J2=Y2*6.0                                                         WNDP3010
      RETURN                                                            WNDP3020
      END                                                               WNDP3030
      SUBROUTINE BLNK(NUM,PLOT,K,*)                                     WNDP3040
C PUTS BLANKS IN PLOT VECTOR LOCATIONS FOR WHICH COUNT IS LESS THAN     WNDP3050
C THE GIVEN VALUE                                                       WNDP3060
      INTEGER*2 NUM(79)                                                 WNDP3070
      LOGICAL*1 PLOT(79),BL/' '/,NB                                     WNDP3080
      NB=.FALSE.                                                        WNDP3090
      DO 100 I=1,79                                                     WNDP3100
      IF(NUM(I)-K)60,80,80                                              WNDP3110
   60 PLOT(I)=BL                                                        WNDP3120
      GOTO 100                                                          WNDP3130
   80 NB=.TRUE.                                                         WNDP3140
  100 CONTINUE                                                          WNDP3150
      IF(NB)RETURN                                                      WNDP3160
      RETURN1                                                           WNDP3170
      END                                                               WNDP3180
```


start
initialize plot (PLØT 0)
read format and key parameter
A
key?
Yes
No
rewind data file
read key
read title
initialize counters, sums, etc.
read limit
Zero plot matrix (PLØT 1)
read next 100 data points
No
key?
Yes
read next 100 data points using key
any points in batch?
No
Yes
store data in temporary file
increment plot matrix (PLØT 2)
take next data point; increment point count and total weight
limit = 0 OR trend in range?
Yes
No
reverse direction
increment sums
more points in batch?
Yes
No
100 points in batch?
Yes
No
B

Flow Diagram for Program WNDPLOT3 - Part. 1


PLOT 0
enter
read direction cosines of counting locations
read parameters
compute information about counting circle
return
PLOT 1
enter
zero count matrix
initialize total weight, total number of data points
return
PLOT 2
enter
take direction cosines and weight of next data point
increment total weight and total number of data points
determine submatrix of plot to be scanned
scan plot submatrix incrementing counters as appropriate
second scan ?
Yes
No
is point near periphery?
No
Yes
more data points ?
Yes
No
return

Flow Diagram for Program WNDPLOT3 - Part 2


PLØT 3
enter
first call?
No
Yes
print contour values and characters
write title
compute density values and set up characters for next line of plot
print line
K ← 5
clear flag
is next density value < K ?
No
Yes
set character to blank
set flag
end of line?
No
Yes
is flag set?
No
Yes
print line
K ← K + 5
K < 95?
Yes
No
end of plot?
No
Yes
return

Flow Diagram for Program WNDPLOT3 - Part 8

Appendix 2

Documentation of Subroutine Library and System Subroutines

Subroutine ... Page

ANDC 383

BSTATC 385

BSTATG 387

CIRC 389

DCAN 391

DCPM1 393

ESTAT1 394

FSTAT1 396

INCH 398

KUIP2 399

ORDERH 400

READK 401

ROTMAT 402

RTSP 403

RTXYEQ 405

SCOM1 407

SRF 409

SYMM 410

TAB1 412

TPXYEQ 413

VRM1 415

WRITEK 416

XKF 417

XK1 419
XK3 420

System Subroutines

PLOTS 421
PLCT 421
SYMBCL 421
NUMBER 421
POLAR 421
URAND 421
WHERE 421

```
C
C ......................................................................
C
C SUBROUTINE ANDC
C
C PURPOSE
C    CONVERSION FROM SPHERICAL TO RECTANGULAR COORDINATES
C    FOR POINTS ON THE UNIT SPHERE.
C
C AUTHOR
C    JOHN RAMSDEN   DEPT OF GEOLOGY   UNIVERSITY OF ALBERTA
C
C DATE WRITTEN
C    JUNE 1973
C
C USAGE
C    CALL ANDC(T,P,J1,J2,A,B,C,J3,N)
C
C DESCRIPTION OF PARAMETERS
C T  VECTOR OF HORIZONTAL ANGLES
C P  VECTOR OF VERTICAL ANGLES
C J1 FIRST INPUT CODE:
C         |J1| = SPACING OF INPUT WORDS IN T AND P
C         J1>0 IF INPUT UNITS ARE DEGREES
C         J2<0 IF INPUT UNITS ARE RADIANS
C    J2  SECOND INPUT CODE:
C         |J2|=1 IF INPUT TYPE IS TREND AND PLUNGE OF LINE
C         |J2|=2 IF INPUT TYPE IS DIP DIRECTION AND DIP OF PLANE
C         |J2|=3 IF INPUT TYPE IS STRIKE AND DIP OF PLANE
C         IN CASES 2 AND 3 THE POLE IS TAKEN AS THE GIVEN POINT
C         J2>0 IF DOMAIN OF INPUT IS THE UNIT SPHERE
C         J2<0 IF DOMAIN OF INPUT IS THE UNIT HEMISPHERE
C    A  VECTOR OF FIRST DIRECTION COSINES
C    B VECTOR OF SECOND DIRECTION COSINES
C    C VECTOR OF THIRD DIRECTION COSINES
C    J3 OUTPUT CODE:
C         |J3|=SPACING OF OUTPUT WORDS IN A, B AND C.
C    N  NUMBER OF POINTS
C
C SUBROUTINES REQUIRED
C    NONE
C
C ......................................................................
C
      SUBROUTINE ANDC(T,P,J1,J2,A,B,C,J3,N)
      REAL T(1),P(1),A(1),B(1),C(1)
      LOGICAL*1 RAD
C
      RAD=.FALSE.
      IF(J1.GT.0)GOTO 100
      RAD=.TRUE.
      J1=-J1
  100 KT=1-J1
      KD=1-J3
      DO 200 I=1,N
      KT=KT+J1
      KD=KD+J3
      TK=T(KT)
      PK=P(KT)
```

```
C                                                                       ANDC 600
C-----------CONVERT ANGLES TO RADIANS                                   ANDC 610
C                                                                       ANDC 620
      IF(RAD)GOTO 150                                                   ANDC 630
      TK=TK*1.745329E-2                                                 ANDC 640
      PK=PK*1.745329E-2                                                 ANDC 650
C                                                                       ANDC 660
C-----------CONVERT ANGLES TO TREND AND PLUNGE                          ANDC 670
C                                                                       ANDC 680
  150 I2=IABS(J2)                                                       ANDC 690
      GOTO (170,165,160),I2                                             ANDC 700
  160 TK=TK+1.570796                                                    ANDC 710
  165 TK=TK+3.141593                                                    ANDC 720
      PK=1.570796-PK                                                    ANDC 730
C                                                                       ANDC 740
C-----------MOVE POINT TO LOWER HEMISPHERE                              ANDC 750
C                                                                       ANDC 760
  170 IF(J2)173,180,180                                                 ANDC 770
  173 IF(PK) 175,180,180                                                ANDC 780
  175 PK=-PK                                                            ANDC 790
      TK=TK+3.141593                                                    ANDC 800
  180 CONTINUE                                                          ANDC 810
C                                                                       ANDC 820
C-----------COMPUTE DIRECTION COSINES                                   ANDC 830
C                                                                       ANDC 840
      CP=COS(PK)                                                        ANDC 850
      A(KD)=COS(TK)*CP                                                  ANDC 860
      B(KD)=SIN(TK)*CP                                                  ANDC 870
      C(KD)=SIN(PK)                                                     ANDC 880
  200 CONTINUE                                                          ANDC 890
      RETURN                                                            ANDC 900
      END                                                               ANDC 910
```

```
C
C ...................................................................
C
C SUBROUTINE BSTATC
C
C PURPOSE
C     TO COMPUTE THE CONCENTRATION PARAMETER ASSUMING DIMROTH-WATSON'S
C     DISTRIBUTION (BIPOLAR CASE) AND CONFIDENCE RADII ON THE ESTIMATED
C     MEAN ASSUMING BINGHAM'S DISTRIBUTION (BIPOLAR CASE).
C
C AUTHOR
C     JOHN RAMSDEN, DEPARTMENT OF GEOLOGY, UNIVERSITY OF ALBERTA,
C     EDMONTON, ALBERTA, CANADA.
C
C DATE
C     1 JULY 1974
C
C USAGE
C     CALL BSTATC(U1,U2,U3,N,XK,R,KO)
C
C DESCRIPTION OF PARAMETERS
C     U1,U2,U3 EIGENVALUES IN INCREASING ORDER OF THE MATRIX U=T/TR(T)
C              WHERE TR(T) IS THE TRACE OF T AND T IS THE CROSS PRODUCT
C              MATRIX OF THE DATA.
C     N        NUMBER OF DATA POINTS
C     XK       ESTIMATED CONCENTRATION PARAMETER
C     R        2*2 MATRIX OF CONFIDENCE RADII; 95% IN COLUMN 1, 99% IN
C              COLUMN 2; RADII TOWARD V1 IN ROW 1, TOWARD V2 IN ROW 2
C     KO       PRINTOUT PARAMETER:
C              KO<0 NO PRINTOUT
C              KO>=0 RESULTS PRINTED ON I/O UNIT KO.
C
C METHOD
C     THE CONCENTRATION PARAMETER IS ESTIMATED BY THE FUNCTION "XK3".
C     THE FORMULA FOR THE CONFIDENCE RADII IS TAKEN FROM MARDIA, K. V.,
C     1972. STATISTICS OF DIRECTIONAL DATA. LONDON & NEW YORK, ACADEMIC
C     PRESS. P.281.  THE SUBROUTINE CONTAINS A TABLE OF THE 95% AND 99%
C     POINTS OF THE F-DISTRIBUTION WITH 2,K DEGREES OF FREEDOM, WHICH IS
C     INTERPOLATED BY THE FUNCTION "TAB1".
C
C OUTPUT
C     IF KO IS NON-NEGATIVE THE SUBROUTINE WRITES THE ESTIMATED
C     CONCENTRATION PARAMETER AND THE CONFIDENCE RADII.
C
C SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C     FUNCTIONS "XK3" AND "TAB1" FROM JR SUBROUTINE LIBRARY.
C
C ...................................................................
C
      SUBROUTINE BSTATC(U1,U2,U3,N,XK,R,KO)
      REAL RTD/57.29578/,NDF(34)/1.,2.,3.,4.,5.,6.,7.,8.,9.,10.
     *,11.,12.,13.,14.,15.,16.,17.,18.,19.,20.,21.,22.,23.,24.,
     *25.,26.,27.,28.,29.,30.,40.,60.,120.,1.0E75/
     *,FD5(34)/199.5,19.0,9.55,6.94,5.79,5.14,4.74,4.46,4.26,4.1,3.98,
     *3.89,3.81,3.74,3.68,3.63,3.59,3.55,3.52,3.49,3.47,3.44,3.42,3.4,
     *3.39,3.37,3.35,3.34,3.33,3.32,3.23,3.15,3.07,3.00/
     *,FD1(34)/4999.5,99.0,30.82,18.0,13.27,10.92,9.55,8.65,8.02,7.56,
     *7.21,6.93,6.70,6.51,6.36,6.23,6.11,6.01,5.93,5.85,5.78,5.72,5.66,
     *5.61,5.57,5.53,5.49,5.45,5.42,5.39,5.18,4.98,4.79,4.61/
     *,N2
C
C
      REAL R(4)
      XK=XK1(U3)
C
      N2=N-2
      F=FAB1(N2,NDF,FD5,34)
      R=U3*(1.0-2.0*F/N2)
      CST=(R-U1)/(U3-U1)
      R(1)=0.0
      IF(CST.GE.0.0 .AND. CST.LT.1.0) R(1)=RTD*ARCOS(SQRT(CST))
      CST=(R-U2)/(U3-U2)
      R(2)=0.0
      IF(CST.GE.0.0 .AND. CST.LT.1.0) R(2)=RTD*ARCOS(SQRT(CST))
C
      F=FAB1(N2,NDF,FD1,34)
      R=U3*(1.0-2.0*F/N2)
      CST=(R-U1)/(U3-U1)
      R(3)=0.0
      IF(CST.GE.0.0 .AND. CST.LT.1.0) R(3)=RTD*ARCOS(SQRT(CST))
      CST=(R-U2)/(U3-U2)
      R(4)=0.0
      IF(CST.GE.0.0 .AND. CST.LT.1.0) R(4)=RTD*ARCOS(SQRT(CST))
      IF(KO.LT.0) RETURN
      WRITE(KO,1) XK,R
    1 FORMAT(' '/' '/'      K FOR              CONFIDENCE RADII'/
     *'     CLUSTER            95%           99%'/
     *1X,F10.3,6X,2(F7.1,F5.1))
      RETURN
      END
```

```
C
C ..............................................................
C
C
C SUBROUTINE BSTATG
C
C PURPOSE
C      TO COMPUTE THE CONCENTRATION PARAMETER ASSUMING DIMROTH-WATSON'S
C      DISTRIBUTION (GIRDLE CASE) AND CONFIDENCE RADII ASSUMING
C      BINGHAM'S DISTRIBUTION (GIRDLE CASE).
C AUTHOR
C      JOHN RAMSDEN, DEPARTMENT OF GEOLOGY, UNIVERSITY OF ALBERTA,
C      EDMONTON, ALBERTA, CANADA.
C
C DATE
C      1 JULY 1974
C
C USAGE
C      CALL BSTATG(U1,U2,U3,N,XK,R,KO)
C
C DESCRIPTION OF PARAMETERS
C      U1,U2,U3 EIGENVALUES IN INCREASING ORDER OF THE MATRIX U=T/TR(T)
C               WHERE TR(T) IS THE TRACE OF T AND T IS THE CROSS PRODUCT
C               MATRIX OF THE DATA.
C      N        NUMBER OF DATA POINTS
C      XK       ESTIMATED CONCENTRATION PARAMETER
C      R        2*2 MATRIX OF CONFIDENCE RADII: 95% IN COLUMN 1,
C               99% IN COLUMN 2; RADII TOWARD V2 IN ROW 1, TOWARD
C               V3 IN ROW 2.
C      KO       PRINTOUT PARAMETER:
C               KO<0   NO PRINTOUT
C               KO>=0  RESULTS PRINTED ON I/O UNIT KO.
C
C METHOD
C      THE CONCENTRATION PARAMETER IS COMPUTED BY FUNCTION "XK1".
C      THE FORMULA FOR COMPUTING THE CONFIDENCE RADII IS TAKEN FROM
C      MARDIA, K. V., 1972. STATISTICS OF DIRECTIONAL DATA. LONDON &
C      NEW YORK, ACADEMIC PRESS. P.279.  THE SUBROUTINE CONTAINS A
C      TABLE OF THE 95% AND 99% POINTS OF THE F-DISTRIBUTION WITH 2,K
C      DEGREES OF FREEDOM, WHICH IS INTERPOLATED BY THE FUNCTION "TAB1".
C
C SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C      FUNCTIONS "XK1" AND "TAB1" FROM JR SUBROUTINE LIBRARY.
C
C ..............................................................
C
      SUBROUTINE BSTATG(U1,U2,U3,N,XK,R,KO)
      REAL RTD/57.29578/,NDF(34)/1.,2.,3.,4.,5.,6.,7.,8.,9.,10.
     *,11.,12.,13.,14.,15.,16.,17.,18.,19.,20.,21.,22.,23.,24.,
     *25.,26.,27.,28.,29.,30.,40.,60.,120.,1.0E75/
     *,FD5(34)/199.5,19.0,9.55,6.94,5.79,5.14,4.74,4.46,4.26,4.1,3.98,
     *3.89,3.81,3.74,3.68,3.63,3.59,3.55,3.52,3.49,3.47,3.44,3.42,3.4,
     *3.39,3.37,3.35,3.34,3.33,3.32,3.23,3.15,3.07,3.00/
     *,FD1(34)/4999.5,99.0,30.82,18.0,13.27,10.92,9.55,8.65,8.02,7.56,
     *7.21,6.93,6.70,6.51,6.36,6.23,6.11,6.01,5.93,5.85,5.78,5.72,5.66,
     *5.61,5.57,5.53,5.49,5.45,5.42,5.39,5.18,4.98,4.79,4.61/
     *,N2
C
      REAL R(4)
      XK=XK1(U1)                                                        BSTG 600
C                                                                       BSTG 610
      N2=N-2                                                            BSTG 620
      F=FAB1(N2,NDF,FD5,34)                                             BSTG 630
      B=U1*(1.0+2.0*F/N2)                                               BSTG 640
      CST=(B-U2)/(U1-U2)                                                BSTG 650
      R(1)=0.0                                                          BSTG 660
      IF(CST.GE.0.0 .AND. CST.LT.1.0) R(1)=RTD*ARCOS(SQRT(CST))         BSTG 670
      CST=(B-U3)/(U1-U3)                                                BSTG 680
      R(2)=0.0                                                          BSTG 690
      IF(CST.GE.0.0 .AND. CST.LT.1.0) R(2)=RTD*ARCOS(SQRT(CST))         BSTG 700
C                                                                       BSTG 710
      F=FAB1(N2,NDF,FD1,34)                                             BSTG 720
      B=U1*(1.0+2.0*F/N2)                                               BSTG 730
      CST=(B-U2)/(U1-U2)                                                BSTG 740
      R(3)=0.0                                                          BSTG 750
      IF(CST.GE.0.0 .AND. CST.LT.1.0) R(3)=RTD*ARCOS(SQRT(CST))         BSTG 760
      CST=(B-U3)/(U1-U3)                                                BSTG 770
      R(4)=0.0                                                          BSTG 780
      IF(CST.GE.0.0 .AND. CST.LT.1.0) R(4)=RTD*ARCOS(SQRT(CST))         BSTG 790
      IF(KO.LT.1) RETURN                                                BSTG 800
      WRITE(KO,1) XK,R                                                  BSTG 810
    1 FORMAT(' '/' '/'      K FOR              CONFIDENCE RADII'/       BSTG 820
     *'     GIRDLE          95%    99%'/                                BSTG 830
     *1X,F10.3,6X,2(F7.1,F5.1))                                         BSTG 840
      RETURN                                                            BSTG 850
      END                                                               BSTG 860
```

```
C                                                                       CIRC  10
C ......................................................................CIRC  20
C                                                                       CIRC  30
C                                                                       CIRC  40
C SUBROUTINE: CIRC1                                                     CIRC  50
C                                                                       CIRC  60
C PURPOSE: TO PERFORM A TEST OF AXIAL SYMMETRY ABOUT A GIVEN AXIS USING CIRC  70
C           KUIPER'S STATISTIC.                                         CIRC  80
C                                                                       CIRC  90
C AUTHOR                                                                CIRC 100
C      JOHN RAMSDEN, DEPARTMENT OF GEOLOGY, UNIVERSITY OF ALBERTA,      CIRC 110
C      EDMONTON, ALBERTA, CANADA.                                       CIRC 120
C                                                                       CIRC 130
C DATE                                                                  CIRC 140
C      1 JULY 1974                                                      CIRC 150
C                                                                       CIRC 160
C USAGE:                                                                CIRC 170
C       CALL CIRC1(DC,W,N,SW,E,J,K,IPRINT)                              CIRC 180
C                                                                       CIRC 190
C PARAMETERS:                                                           CIRC 200
C      DC  3*N MATRIX OF DIRECTION COSINES                              CIRC 210
C      W   VECTOR OF WEIGHTS                                            CIRC 220
C      N   NUMBER OF DATA POINTS                                        CIRC 230
C      SW  SUM OF WEIGHTS                                               CIRC 240
C      E   VECTOR OF LENGTH 2 OR 3 - SEE J                              CIRC 250
C      J   INPUT PARAMETER:                                             CIRC 260
C          =1 IF E CONTAINS TREND AND PLUNGE OF AXIS IN DEGREES         CIRC 270
C          =2 IF E CONTAINS TREND AND PLUNGE OF AXIS IN RADIANS         CIRC 280
C          =3 IF E CONTAINS DIRECTION COSINES OF AXIS                   CIRC 290
C      K   TEST RESULT:                                                 CIRC 300
C          =0 IF TEST NOT SIGNIFICANT                                   CIRC 310
C          =1 IF TEST SIGNIFICANT (UNIFORMITY REJECTED)                 CIRC 320
C      IPRINT PRINTOUT PARAMETER:                                       CIRC 330
C             <0  NO PRINTOUT                                           CIRC 340
C             >=0 RESULTS PRINTED ON I/O UNIT IPRINT.                   CIRC 350
C                                                                       CIRC 360
C METHOD                                                                CIRC 370
C      SUBROUTINE "VRM1" IS CALLED TO OBTAIN THE MATRIX THAT WILL EFFECT CIRC 380
C      THE ROTATION THAT WILL BRING THE GIVIN AXIS INTO A VERTICAL      CIRC 390
C      POSITION.  SUBROUTINE "GMPRD" IS CALLED TO PERFOM THE MATRIX     CIRC 400
C      MULTIPLICATION.  SUBROUTINE "RTSP" COMPUTES THE NEW TRENDS AND   CIRC 410
C      PLUNGES.  FROM THE TRENDS THE SUBROUTINE COMPUTES A WEIGHTED     CIRC 420
C      FREQUENCY VECTOR WITH 360 1-DEGREE CLASSES.  THIS IS PASSED TO   CIRC 430
C      SUBROUTINE "KUIP2" WHICH PERFORMS KUIPER'S TEST FOR UNIFORMITY   CIRC 440
C      ON THE CIRCLE.                                                   CIRC 450
C                                                                       CIRC 460
C SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                         CIRC 470
C      SUBROUTINES "VRM1", "ROTMAT", "RTSP" AND "KUIP2" FROM            CIRC 480
C      JR SUBROUTINE LIBRARY.                                           CIRC 490
C      SUBROUTINES "GMPRD" AND "GTPRD" FROM SSP SUBROUTINE LIBRARY.     CIRC 500
C                                                                       CIRC 510
C ......................................................................CIRC 520
C                                                                       CIRC 530
      SUBROUTINE CIRC1(DC,DCR,W,N,SW,E,J,K,IPRINT)                      CIRC 540
      REAL DC(3,1),W(1),E(3),DCR(3,1),F(360)                            CIRC 550
      REAL RM(9)                                                        CIRC 560
C                                                                       CIRC 570
      CALL VRM1(E,RM,J)                                                 CIRC 580
      CALL GMPRD(RM,DC,DCR,3,3,N)                                       CIRC 590
      CALL PTSP(DCR,DCR(2,1),DCR(3,1),3,DCR,DCR(2,1),DCR(3,1),3,1,N)    CIRC 600
      DO 320 I=1,360                                                    CIRC 610
  320 F(I)=0.0                                                          CIRC 620
      DO 355 I=1,N                                                      CIRC 630
      L=DCR(1,I)+1.0                                                    CIRC 640
  355 F(L)=F(L)+W(I)                                                    CIRC 650
      FACT=N/SW                                                         CIRC 660
      DO 380 I=1,360                                                    CIRC 670
  380 F(I)=F(I)*FACT                                                    CIRC 680
      CALL KUIP2(F,N,K,IPRINT)                                          CIRC 690
      RETURN                                                            CIRC 700
      END                                                               CIRC 710
```

```
C
C .....................................................................
C
C
C
C SUBROUTINE:    DCAN
C
C PURPOSE:    TO CONVERT FROM RECTANGULAR TO SPHERICAL COORDINATES
C
C AUTHOR:     JOHN RAMSDEN    DEPT OF GEOLOGY    UNIVERSITY OF ALBERTA
C             EDMONTON    ALBERTA
C
C DATE WRITTEN :     JUNE 1973
C
C USAGE:            CALL DCAN(A,B,G,J1,T,P,J2,J3,N)
C
C DESCRIPTION OF PARAMETERS:
C    A     VECTOR OF FIRST DIRECTION COSINES
C    B     VECTOR OF SECOND DIRECTION COSINES
C    G     VECTOR OF THIRD DIRECTION COSINES
C    J1    INPUT DESCRIPTION PARAMETER:
C          |J1|= SPACING OF DIRECTION COSINES IN A, B, AND G IN WORDS
C    T     VECTOR OF HORIZONTAL ANGLES
C    P     VECTOR OF VERTICAL ANGLES
C    J2    FIRST OUTPUT DESCRIPTION PARAMETER:
C          |J2|= SPACING OF OUTPUT WORDS IN T AND P
C           J2 >0 IF OUTPUT UNITS TO BE DEGREES
C           J2 <0 IF OUTPUT UNITS TO BE RADIANS
C    J3    SECOND OUTPUT DESCRIPTION PARAMETER:
C          |J3|=1 IF OUTPUT TYPE TO BE TREND AND PLUNGE
C          |J3|=2 IF OUTPUT TYPE TO BE DIP DIRECTION AND DIP
C          |J3|=3 IF OUTPUT TYPE TO BE STRIKE AND DIP
C           J3 >0 IF DOMAIN IS THE SPHERE
C           J3 <0 IF DOMAIN IS THE HEMISPHERE
C    N     NUMBER OF DATA POINTS
C
C
C .....................................................................
C
      SUBROUTINE DCAN(A,B,G,J1,T,P,J2,J3,N)
      REAL A(N),B(N),G(N),T(N),P(N),RTD/57.29578/
      IF(N.LT.1)RETURN
      I1=IABS(J1)
      I2=IABS(J2)
      K1=1-I1
      K2=1-I2
      DO 100 I=1,N
      K1=K1+I1
      K2=K2+I2
   10 AA=A(K1)
      BB=B(K1)
      GG=G(K1)
C                     ADJUST OUTPUT TO REQUIRED DOMAIN
      IF(J3) 13,13,15
   13 IF(GG) 14,15,15
   14 AA=-AA
      BB=-BB
      GG=-GG
   15 PL=ARSIN(GG)
      IF(AA) 18,20,18
   18 TR=ATAN(BB/AA)
      IF(TR.LT.0.0) TR=TR+3.14159265
      IF(BB.EQ.0.0 .AND. AA.LT.0.0) TR=3.14159265
      GOTO 30
   20 TR=1.57079633
   30 IF(BB.LT.0.0) TR=TR+3.14159265
C                           ADJUST OUTPUT TO REQUIRED TYPE
      I3=IABS(J3)
      GOTO (38,36,34),I3
   34 TR=TR-1.570796
   36 TR=TR-3.141593
      IF(TR.LT.0.0) TR=TR+6.283185
      PL=1.570796-PL
C                           ADJUST OUTPUT TO REQUIRED UNITS
   38 IF(J2) 40,40,50
   40 T(K2)=TR
      P(K2)=PL
      GOTO 100
   50 T(K2)=TR*RTD
      P(K2)=PL*RTD
  100 CONTINUE
      RETURN
      END
```

```
C
C.......................................................................
C
C SUBROUTINE: *DCPM1
C
C PURPOSE:  TO COMPUTE THE MATRIX OF SQUARES AND PRODUCTS OF DIRECTION
C     COSINES ("CROSS PRODUCT MATRIX") FOR A SET OF DIRECTIONS.
C
C AUTHOR
C     JOHN RAMSDEN, DEPARTMENT OF GEOLOGY, UNIVERSITY OF ALBERTA,
C     EDMONTON, ALBERTA, CANADA.
C
C DATE
C     1 JULY 1974
C
C USAGE:
C      CALL DCPM1(DC,W,N,U)
C
C PARAMETERS:
C     DC 3*N MATRIX OF DIRECTION COSINES OF DATA POINTS
C     W  VECTOR OF WEIGHTS
C     N  NUMBER OF DATA POINTS
C     U  DATA CROSS-PRODUCT MATRIX (SYMMETRIC STORAGE MODE)
C
C SUBROUTINES CALLED:  NONE
C
C.......................................................................
C
      SUBROUTINE DCPM1(DC,W,N,U)
      REAL DC(3,1),W(1),U(6)
C
      DO 100 I=1,6
  100 U(I)=0.0
C
      DO 200 J=1,N
      A=DC(1,J)
      B=DC(2,J)
      G=DC(3,J)
      WJ=W(J)
      WA=WJ*A
      WB=WJ*B
      WG=WJ*G
      U(1)=U(1)+WA*A
      U(2)=U(2)+WA*B
      U(3)=U(3)+WB*B
      U(4)=U(4)+WA*G
      U(5)=U(5)+WB*G
      U(6)=U(6)+WG*G
  200 CONTINUE
      RETURN
C     DEBUG SUBTRACE,SUBCHK
      END
```

```
C
C .....................................................................
C
C SUBROUTINE: ESTAT1
C
C PURPOSE
C     TO OBTAIN THE EIGENVALUES AND EIGENVECTORS OF THE DATA CROSS
C     PRODUCT MATRIX, COMPUTE ANGULAR COORDINATES FOR THE EIGENVECTORS,
C     PERFORM BINGHAM'S TESTS OF SYMMETRY USING THE EIGENVALUES, AND
C     COMPUTE STATISTICS FOR AXIALLY SYMMETRIC CASES.
C
C
C AUTHOR
C     JOHN RAMSDEN, DEPARTMENT OF GEOLOGY, UNIVERSITY OF ALBERTA,
C     EDMONTON, ALBERTA, CANADA.
C
C DATE
C     1 JULY 1974
C
C USAGE
C     CALL ESTAT1(N,TITLE,SW,U,E,S,T,IR,XK,R,IPRINT)
C
C DESCRIPTION OF PARAMETERS
C     N      NUMBER OF DATA POINTS
C     TITLE  120-BYTE TITLE
C     SW     SUM OF WEIGHTS OF DATA POINTS
C     U      DATA CROSS PRODUCT MATRIX IN SYMMETRIC STORAGE MODE
C     E      MATRIX OF EIGENVECTORS.  ONE EIGENVECTOR IN EACH COLUMN,
C            IN ORDER OF INCREASING EIGENVALUES.
C     S      MATRIX OF ANGULAR COORDINATES OF EIGENVECTORS.  COLUMNS
C            CORRESPOND TO COLUMNS OF E.  ROWS 1 AND 2 CONTAIN TREND
C            AND PLUNGE; ROWS 3 AND 4 CONTAIN DIP DIRECTION AND DIP
C            OF PERPENDICULAR PLANE.
C     T      MATRIX OF EIGENVALUES.  COLUMNS ARE IN ORDER OF INCREASING
C            EIGENVALUES.  ROW 1 CONTAINS EIGENVALUES; ROW 2 CONTAINS
C            EIGENVALUES SCALED SO THAT THEIR SUM IS 1.
C     IR     RESULT OF SYMMETRY TESTS (SEE SUBROUTINE "SYMM")
C     XK     CONCENTRATION PARAMETER OF DIMROTH-WATSON'S DISTRIBUTION
C            FOR THE AXIALLY SYMMETRIC CASES.
C     R      MATRIX OF CONFIDENCE RADII ON MEAN FOR AXIALLY SYMMETRIC
C            CASES (SEE SUBROUTINES "BSTATC" AND "BSTATG").
C     IPRINT PRINTOUT PARAMETER:
C            IPRINT<0  NO PRINTOUT
C            IPRINT>=0 RESULTS PRINTED ON I/O UNIT "IPRINT".
C
C METHOD
C     SUBROUTINE "EIGEN" IS CALLED TO COMPUTE THE EIGENVALUES AND
C     EIGENVECTORS; SUBROUTINE "RTSP" COMPUTES THE SPHERICAL COORDINATES
C     OF THE EIGENVECTORS. SUBROUTINE "SYMM" PERFORMS THE TESTS OF
C     SYMMETRY, WHILE SUBROUTINES "BSTATC" AND "BSTATG" COMPUTE
C     STATISTICS FOR THE AXIALLY SYMMETRIC CASES.
C
C SUBROUTINES REQUIRED
C     SUBROUTINES "RTSP", "SYMM", "BSTATC", "BSTATG" AND FUNCTIONS
C     "XK1", "XK3" AND "TAB1" FROM JR SUBROUTINE LIBRARY;
C     SUBROUTINE "EIGEN" FROM SSP SUBROUTINE LIBRARY.
C
C .....................................................................
C
```

```
      SUBROUTINE ESTAT1(N,TITLE,SW,U,E,S,T,IR,XK,R,IPRINT)
      INTEGER TITLE(30)
      REAL U(6),T(6),E(3,3),S(12),R(4),TT(2,3),SS(4,3),POLR(12)
C
      WRITE(IPRINT,1)TITLE,N,SW
    1 FORMAT(' ',30A4/1X,I5,' POINTS WITH TOTAL WEIGHT OF',F7.1)
      WRITE(IPRINT,2)
    2 FORMAT(' '/' "BINGHAM" STATISTICS')
C
      DO 230 I=1,6
  230 T(I)=U(I)
      CALL EIGEN(T,E,3,0)
      T(5)=T(1)
      T(1)=T(6)
      TRACE=T(1)+T(3)+T(5)
      T(2)=T(1)/TRACE
      T(4)=T(3)/TRACE
      T(6)=T(5)/TRACE
      DO 270 I=1,3
      A=E(I,1)
      E(I,1)=E(I,3)
  270 E(I,3)=A
C
C-----COMPUTE ANGLES
C
      CALL RTSP(E(1,1),E(2,1),E(3,1),3,S(1),S(2),POLR,4,-1,3)
      CALL RTSP(E(1,1),E(2,1),E(3,1),3,S(3),S(4),POLR,4,-2,3)
      IF(IPRINT.LT.0)GOTO310
      L=0
      DO 280 J=1,3
      DO 280 I=1,4
      L=L+1
  280 SS(I,J)=S(L)
      L=0
      DO 290 J=1,3
      DO 290 I=1,2
      L=L+1
  290 TT(I,J)=T(L)
      WRITE(IPRINT,4)
      DO 300 J=1,3
  300 WRITE(IPRINT,5) (TT(I,J),I=1,2), (E(I,J),I=1,3), (SS(I,J),I=1,4)
  310 CONTINUE
C
    4 FORMAT(' '/' '/1X,' EIGENVAL   EVAL/SW    DIRECTION   COSINES
     *  TR   PL     DD    DP')
    5 FORMAT(' '/1X,F9.2,4F9.5,2(F7.1,F5.1))
      CALL SYMM(T(2),T(4),T(6),N,IR,IPRINT)
      IF(IR.EQ.1)CALL BSTATG(T(2),T(4),T(6),N,XK,R,IPRINT)
      IF(IR.EQ.2)CALL BSTATC(T(2),T(4),T(6),N,XK,R,IPRINT)
C
      RETURN
C     DEBUG SUBTRACE,SUBCHK
      END
```

```
C
C.................................................................
C
C SUBROUTINE FSTAT1
C
C PURPOSE
C      GIVEN THE SUMS OF COMPONENTS OF A SET OF VECTORS, TO COMPUTE THE
C      LENGTH, DIRECTION COSINES AND ANGULAR COORDINATES OF THE
C      RESULTANT, AND TO COMPUTE THE CONCENTRATION PARAMETER AND
C      CONFIDENCE RADIUS ON THE MEAN ASSUMING FISHER'S DISTRIBUTION.
C
C AUTHOR
C      JOHN RAMSDEN, DEPARTMENT OF GEOLOGY, UNIVERSITY OF ALBERTA,
C      EDMONTON, ALBERTA, CANADA.
C
C DATE
C      1 JULY 1974
C
C USAGE
C      CALL FSTAT1(TITLE,N,SW,U,E,S,T,XK,CR,IPRINT)
C
C DESCRIPTION OF PARAMETERS
C      TITLE  120-BYTE TITLE
C      N      NUMBER OF DATA POINTS
C      SW     SUM OF WEIGHTS OF DATA POINTS
C      U      VECTOR OF SUMS OF COMPONENTS
C      E      VECTOR OF DIRECTION COSINES OF RESULTANT
C      S      VECTOR OF ANGULAR COORDINATES OF RESULTANT CONTAINING
C             TREND AND PLUNGE FOLLOWED BY THE DIP DIRECTION AND DIP OF
C             THE PERPENDICULAR PLANE.
C      T      VECTOR CONTAINING (1) LENGTH OF RESULTANT AND (2) LENGTH
C             OF RESULTANT DIVIDED BY SW.
C      XK     ESTIMATED CONCENTRATION PARAMETER OF POPULATION
C      CR     CONFIDENCE RADII ON MEAN: 95% AND 99%.
C      IPRINT PRINTOUT PARAMETER:
C             IPRINT<0  NO PRINTOUT
C             IPRINT>=0 RESULTS PRINTED ON I/O UNIT "IPRINT".
C
C METHOD
C      SUBROUTINE "RTSP" COMPUTES THE ANGULAR COORDINATES OF THE
C      RESULTANT. FUNCTION "XKF" ESTIMATES THE CONCENTRATION PARAMETER
C      OF THE POPULATION. THE CONFIDENCE RADII ARE COMPUTED USING A
C      FORMULA GIVEN BY WATSON, G. S., 1957, STATISTICAL METHODS IN ROCK
C      MAGNETISM, MONTHLY NOTICES, ROYAL ASTRONOMICAL SOCIETY, GEOPHYS.
C      SUPPL., P.292.
C
C SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C      SUBROUTINES "RTSP" AND "XKF" FROM JR SUBROUTINE LIBRARY.
C
C .................................................................
C
      SUBROUTINE FSTAT1(TITLE,N,SW,U,E,S,T,XK,CR,IPRINT)
      INTEGER TITLE(30)
      REAL U(3),T(2),E(3),S(4),CR(2),RTD/57.29578/
C
   10 IF(IPRINT.GE.0) WRITE(IPRINT,1) TITLE,N,SW
    1 FORMAT(' ',30A4/1X,I5,' POINTS WITH TOTAL WEIGHT OF',F7.1)
   20 IF(IPRINT.GE.0) WRITE(IPRINT,2)
    2 FORMAT(' '/' "FISHER" STATISTICS')
```

```
 30 X=U(1)                                                               FSTA 600
 40 Y=U(2)                                                               FSTA 610
 50 Z=U(3)                                                               FSTA 620
 60 R=SQRT(X*X+Y*Y+Z*Z)                                                  FSTA 630
 70 A=X/R                                                                FSTA 640
 80 B=Y/R                                                                FSTA 650
 90 G=Z/R                                                                FSTA 660
100 IF(ABS(A).GT.0.999999) A=SIGN(0.999999,A)                            FSTA 670
110 IF(ABS(B).GT.0.999999) B=SIGN(0.999999,B)                            FSTA 680
120 IF(ABS(G).GT.0.999999) G=SIGN(0.999999,G)                            FSTA 690
130 CALL RTSP(A,B,G,1,S(1),S(2),POLR,1,1,1)                              FSTA 700
140 CALL RTSP(A,B,G,1,S(3),S(4),POLR,1,2,1)                              FSTA 710
150 T(1)=R                                                               FSTA 720
160 T(2)=R/SW                                                            FSTA 730
170 E(1)=A                                                               FSTA 740
180 E(2)=B                                                               FSTA 750
190 E(3)=G                                                               FSTA 760
200 IF(IPRINT.GE.0) WRITE(IPRINT,3)T,E,S                                 FSTA 770
  3 FORMAT(' '/'        R         R/SW       DIRECTION    COSINES',      FSTA 780
   *'        TR    PL       DD    DP'/                                   FSTA 790
   *1X,F9.2,4F9.5,2(F7.1,F5.1))                                          FSTA 800
210 RB=R/SW                                                              FSTA 810
    R=RB*N                                                               FSTA 820
    CALL XKF(RB,N,XK,IER)                                                FSTA 830
230 EX=1.0/(N-1.0)                                                       FSTA 840
240 Q=(N-R)/R                                                            FSTA 850
250 CR(1)=RTD*ARCOS(1.0-(20.0**EX-1.0)*Q)                                FSTA 860
260 CR(2)=RTD*ARCOS(1.0-(100.**EX-1.0)*Q)                                FSTA 870
270 IF(IPRINT.LT.0) RETURN                                               FSTA 880
280 WRITE(IPRINT,4) XK,CR(1),CR(2)                                       FSTA 890
  4 FORMAT(' '/'     FISHER     CONFIDENCE RADII'/                       FSTA 900
   *'      K            95%    99%'/                                     FSTA 910
   *F10.3,F10.1,F7.1)                                                    FSTA 920
    RETURN                                                               FSTA 930
C      DEBUG SUBTRACE,SUBCHK                                             FSTA 940
C      AT 10                                                             FSTA 950
C      TRACE ON                                                          FSTA 960
    END                                                                  FSTA 970
```

```
C
C .....................................................................
C
C SUBROUTINE INCH
C
C PURPOSE
C    TO CONVERT AN INTEGER NUMBER TO A CHARACTER STRING
C
C AUTHOR
C    JOHN RAMSDEN   DEPARTMENT OF GEOLOGY    UNIVERSITY OF ALBERTA
C
C DATE WRITTEN
C 27 SEPTEMBER 1974
C
C USAGE
C    CALL INCH(M,S,L,J,FILL,IS,IER)
C
C DESCRIPTION OF PARAMETERS
C M       FULLWORD INTEGER TO BE CONVERTED
C S       LOCATION OF VECTOR INTO WHICH DIGITS ARE TO BE PLACED
C L       NUMBER OF LOCATIONS IN S TO BE USED
C J       SPACING OF CHARACTERS IN S (IN BYTES)
C FILL    A SINGLE BYTE CONTAINING THE CHARACTER TO BE USED
C         TO PAD THE BEGINNING OF THE STRING
C IS      THE LOCATION IN S CONTAINING THE LEFTMOST CHARACTER
C         OF THE NUMBER
C IER     ERROR CODE: IER=1 : NUMBER OF CHARACTERS IN NUMBER
C                             GREATER THAN L
C
C COMMENTS
C    THE DIGITS ARE PLACED IN L RIGHT-JUSTIFIED. ANY UNUSED LOCATIONS
C    AT THE BEGINNING OR L ARE FILLED WITH THE FILL CHARACTER.
C    THE BYTES OF L INTO WHICH DIGITS ARE PLACED ARE 1, 1+J, 1+2J, ETC.
C
C .....................................................................
C
      SUBROUTINE INCH(M,S,L,J,FILL,IS,IER)
      LOGICAL*1 S(1),STAR/'*'/,FILL
     *,C(10)/'0','1','2','3','4','5','6','7','8','9'/
C
      IER=0
      N=M
      I=(L-1)*J+1
   20 K=MOD(N,10)
      S(I)=C(K+1)
      N=N/10
      IF(N.EQ.0)GOTO200
      I=I-J
      IF(I)300,300,20
  200 IS=I/J+1
      IF(J.EQ.1)IS=I
  210 IF(I.EQ.1)RETURN
      I=I-J
      S(I)=FILL
      GOTO210
  300 I=1-J
      DO 310 II=1,L
      I=I+J
  310 S(I)=STAR
      IER=1
      IS=0
      RETURN
      END
```

```
C
C .............................................................
C
C SUBROUTINE: KUIP2
C
C PURPOSE: GIVEN A VECTOR OF FREQUENCIES IN 360 EQUAL CLASSES ON THE
C          CIRCLE, TO COMPUTE KUIPER'S STATISTIC FOR TESTING UNIFORMITY.
C
C
C AUTHOR
C     JOHN RAMSDEN, DEPARTMENT OF GEOLOGY, UNIVERSITY OF ALBERTA,
C     EDMONTON, ALBERTA, CANADA.
C
C DATE
C     1 JULY 1974
C
C USAGE
C     CALL KUIP2(F,N,IR,IO)
C
C PARAMETERS:
C     F VECTOR OF FREQUENCIES
C     N NUMBER OF DATA POINTS IN SAMPLE
C     IR TEST RESULT =0 IF TEST NOT SIGNIFICANT (UNIFORMITY ACCEPTED)
C                    =1 IF TEST SIGNIFICANT (UNIFORMITY REJECTED)
C     IO PRINTOUT CONTROL PARAMETER:
C        IO<0 NO PRINTOUT
C        IO>=0 TEST RESULTS PRINTED ON I/O UNIT IO
C
C METHOD
C     THE SUBROUTINE COMPUTES THE STATISTIC
C                VS = (SN + 0.155 + 0.24/SN) * VN
C     WHERE SN=SQRT(N) AND VN IS THE STATISTIC DEFINED BY KUIPER.  THE
C     COMPUTED VALUE IS COMPARED WITH THE 95% POINT OF THE NULL
C     DISTRIBUTION OF "VS", AND THE RESULT OF THE TEST IS RETURNED IN
C     "IR". THE COMPUTED VALUE OF "VS" IS NOT RETURNED. THE NULL
C     DISTRIBUTION OF "VS" IS INDEPENDENT OF "N". THIS TEST IS VALID
C     ONLY FOR N>8.
C SUBROUTINES CALLED: NONE
C
C
C SEE MARDIA,K.V. 1972 STATISTICS OF DIRECTIONAL DATA. ACADEMIC PRESS.
C     LONDON & NEW YORK. PP 173-180.
C
C .............................................................
C
      SUBROUTINE KUIP2(F,N,IR,IO)
      REAL F(360)
C
      IR=0
      AMIN=1.0E75
      AMAX=-1.0E75
      AN=N
      RN=1.0/AN
      SN=SQRT(AN)
      T=0.0
      E=0.0
      DO 200 I=1,360
      E=E+F(I)*RN
      T=T+2.777778E-3
      D=T-E
      IF(AMIN.GT.D)AMIN=D
      IF(AMAX.LT.D)AMAX=D
  200 CONTINUE
      VN=AMAX-AMIN+RN
      VS=(SN+0.155+0.24/SN)*VN
      IF(VS.GT.1.747)IR=1
      IF(IO.GE.0) WRITE(IO,1) VS
      RETURN
C
    1 FORMAT(' '/' KUIPER''S STATISTIC =',F10.3,5X,'95% POINT = 1.747')
C     DEBUG SUBTRACE,SUBCHK
      END
```

```
C
C ..........................................................................
C
C
C SUBROUTINE ORDERH
C
C PURPOSE
C      TO ORDER A SET OF HALFWORD INTEGERS
C
C AUTHOR
C      JOHN RAMSDEN, DEPARTMENT OF GEOLOGY, UNIVERSITY OF ALBERTA,
C      EDMONTON, ALBERTA, CANADA.
C
C DATE
C      1 SEPTEMBER 1974
C
C USAGE
C     CALL ORDERH(M,N)
C
C PARAMETERS
C     M  VECTOR OF HALFWORD INTEGERS TO BE ORDERED
C     N  LENGTH OF M
C
C ..........................................................................
C
      SUBROUTINE ORDERH(M,N)
      INTEGER*2 M(N),K
      J=1
  100 MIN=99999999
      DO 200 I=J,N
      IF(M(I).GE.MIN)GOTO200
      MIN=M(I)
      IMIN=I
  200 CONTINUE
      K=M(IMIN)
      M(IMIN)=M(J)
      M(J)=K
      J=J+1
      IF(J.LE.N-1)GOTO100
      RETURN
      END
```

```
C
C ..........................................................................
C
C SUBROUTINE READK
C
C PURPOSE
C    TO READ A SET OF INTEGERS FORMATTED BY SUBROUTINE WRITEK
C
C AUTHOR
C    JOHN RAMSDEN   DEPT OF GEOLOGY   UNIVERSITY OF ALBERTA
C
C DATE WRITTEN
C    MARCH 1973
C
C USAGE
C    CALL READK(IOU,TITLE,LENFIL,LENKEY,KEY,*,*)
C
C DESCRIPTION OF PARAMETERS
C    IOU    I/O UNIT TO BE READ
C    TITLE  TITLE OF THE SET READ
C    LENFIL SIZE OF THE UNIVERSE TO WHICH THE SET BELONGS
C    LENKEY SIZE OF SET
C    KEY    SET
C    (RETURN1) RETURN TAKEN IF END-OF-FILE ENCOUNTERED AT FIRST RECORD
C    (RETURN2) RETURN TAKEN IF END-OF-FILE ENCOUNTERED AFTER
C              FIRST RECORD (SET NOT COMPLETE) OF IF RECORDS ARE
C              OUT OF SEQUENCE
C
C SUBROUTINES REQUIRED
C    NONE
C
C
C
C ..........................................................................
C
      SUBROUTINE READK(IOU,TITLE,LENFIL,LENKEY,KEY,*,*)
C
    1 FORMAT(18A4,5X,I3)
    2 FORMAT(12A4,4X,I4,4X,I4,13X,I3)
    3 FORMAT(18I4,5X,I3)
C
      INTEGER TITLE(30)
      INTEGER*2 KEY(5018)
C
C
      READ(IOU,1,END=200) (TITLE(I),I=1,18),N
      IF(N.NE.1)GOTO 220
      READ(IOU,2,END=220) (TITLE(I),I=19,30),LENFIL,LENKEY,N
      IF(N.NE.2)GOTO 220
      IF(LENKEY.EQ.0)RETURN
C
      KA=1
      KB=18
      NC=3
  100 READ(IOU,3,END=220) (KEY(I),I=KA,KB),N
      IF(N.NE.NC)GOTO 220
      KA=KA+18
      IF(KA.GT.LENKEY)RETURN
      KB=KB+18
      NC=NC+1
      GOTO 100
C
  200 RETURN1
  220 RETURN2
C
C
      END
```

```
C
C ..........................................................................
C
C SUBROUTINE ROTMAT
C
C PURPOSE
C     TO COMPUTE THE 3*3 ORTHOGONAL TRANSFORMATION MATRIX T SUCH THAT
C     Y=TX WHERE Y IS THE NEW POSITION OF THE VECTOR X AFTER A
C     SPECIFIED ROTATION ABOUT A SPECIFIED AXIS.
C
C AUTHOR
C     JOHN RAMSDEN, DEPARTMENT OF GEOLOGY, UNIVERSITY OF ALBERTA,
C     EDMONTON, ALBERTA, CANADA.
C
C DATE
C     1 FEBRUARY 1972
C
C USAGE
C     CALL ROTMAT(A,B,C,T)
C
C PARAMETERS
C     A  TREND OF AXIS OF ROTATION IN RADIANS
C     B  PLUNGE OF AXIS OF ROTATION IN RADIANS
C     C  ANGLE OF ROTATION IN RADIANS.  POSITIVE ROTATION OF THE VECTOR
C        X ABOUT THE SPECIFIED AXIS IS CLOCKWISE TO AN OBSERVER AT THE
C        ORIGIN.
C     T  RETURNED 3*3 ROTATION MATRIX.
C
C METHOD
C     THE SUBROUTINE CONSTRUCTS DIRECTLY THE ROTATION MATRICES FOR
C     (1) THE ROTATION THAT WILL BRING THE SPECIFIED AXIS INTO THE
C     POSITIVE X-AXIS, AND (2) THE SPECIFIED ROTATION, PERFORMED ABOUT
C     THE POSITIVE X-AXIS.  IF THESE TWO MATRICES ARE RESPECTIVELY U
C     AND V, THE REQUIRED MATRIX IS GIVEN BY
C                          T = W.V.U
C     WHERE T IS THE REQUIRED MATRIX AND W IS THE TRANSPOSE OF U.
C
C SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED.
C     SUBROUTINES "GMPRD" AND "GTPRD" FROM SSP SUBROUTINE LIBRARY.
C
C
C ..........................................................................
C
      SUBROUTINE ROTMAT(A,B,C,T)
      REAL T(9),P(9),Q(9),R(9)
      SINA=SIN(A)
      COSA=COS(A)
      SINB=SIN(B)
      COSB=COS(B)
      SINC=SIN(C)
      COSC=COS(C)
      Q(1)=COSA*COSB
      Q(2)=-SINA
      Q(3)=-COSA*SINB
      Q(4)=SINA*COSB
      Q(5)=COSA
      Q(6)=-SINA*SINB
      Q(7)=SINB
      Q(8)=0.
      Q(9)=COSB
      P(1)=1.
      P(2)=0.
      P(3)=0.
      P(4)=0.
      P(5)=COSC
      P(6)=SINC
      P(7)=0.
      P(8)=-SINC
      P(9)=COSC
      CALL GMPRD(P,Q,R,3,3,3)
      CALL GTPRD(Q,R,T,3,3,3)
      RETURN
      END
```

```
C
C ......................................................................
C
C
C
C SUBROUTINE:     RTSP
C
C PURPOSE:     TO CONVERT FROM RECTANGULAR TO SPHERICAL COORDINATES
C
C AUTHOR:      JOHN RAMSDEN     DEPT OF GEOLOGY     UNIVERSITY OF ALBERTA
C              EDMONTON    ALBERTA
C
C DATE WRITTEN :     JUNE 1973
C
C USAGE:            CALL RTSP(A,B,G,J1,T,P,R,J2,J3,N)
C
C DESCRIPTION OF PARAMETERS:
C     A     VECTOR OF X COMPONENTS
C     B     VECTOR OF Y COMPONENTS
C     G     VECTOR OF Z COMPONENTS
C     J1    INPUT DESCRIPTION PARAMETER:
C           |J1|= SPACING OF COMPONENTS          IN A, B, AND G IN WORDS
C     T     VECTOR OF HORIZONTAL ANGLES
C     P     VECTOR OF VERTICAL ANGLES
C     R     VECTOR OF DISTANCES FROM ORIGIN
C     J2    FIRST OUTPUT DESCRIPTION PARAMETER:
C           |J2|= SPACING OF OUTPUT WORDS IN T AND P
C            J2 >0 IF OUTPUT UNITS TO BE DEGREES
C            J2 <0 IF OUTPUT UNITS TO BE RADIANS
C     J3    SECOND OUTPUT DESCRIPTION PARAMETER:
C           |J3|=1 IF OUTPUT TYPE TO BE TREND AND PLUNGE
C           |J3|=2 IF OUTPUT TYPE TO BE DIP DIRECTION AND DIP
C           |J3|=3 IF OUTPUT TYPE TO BE STRIKE AND DIP
C            J3 >0 IF DOMAIN IS THE SPHERE
C            J3 <0 IF DOMAIN IS THE HEMISPHERE
C     N     NUMBER OF DATA POINTS
C
C
C
C ......................................................................
C
      SUBROUTINE RTSP(A,B,G,J1,T,P,R,J2,J3,N)
      REAL A(N),B(N),G(N),T(N),P(N),RTD/57.29578/,R(N)
      IF(N.LT.1)RETURN
      I1=IABS(J1)
      I2=IABS(J2)
      K1=1-I1
      K2=1-I2
      DO 100 I=1,N
      K1=K1+I1
      K2=K2+I2
   10 AA=A(K1)
      BB=B(K1)
      GG=G(K1)
C                          ADJUST OUTPUT TO REQUIRED DOMAIN
      IF(J3) 13,13,15
   13 IF(GG) 14,15,15
   14 AA=-AA
      BB=-BB
      GG=-GG                                                            RTSP 600
   15 RO=SQRT(AA*AA+BB*BB+GG*GG)                                        RTSP 610
      R(K2)=RO                                                          RTSP 620
      AA=AA/RO                                                          RTSP 630
      BB=BB/RO                                                          RTSP 640
      GG=GG/RO                                                          RTSP 650
      PL=ARSIN(GG)                                                      RTSP 660
      IF(AA)18,20,18                                                    RTSP 670
   18 TR=ATAN(BB/AA)                                                    RTSP 680
      IF(TR.LT.0.0)TR=TR+3.14159265                                     RTSP 690
      IF(BB.EQ.0.0 .AND. AA.LT.0.0)TR=3.14159265                        RTSP 700
      GOTO 30                                                           RTSP 710
   20 TR=1.57079633                                                     RTSP 720
   30 IF(BB.LT.0.0)TR=TR+3.14159265                                     RTSP 730
C                           ADJUST OUTPUT TO REQUIRED TYPE              RTSP 740
      I3=IABS(J3)                                                       RTSP 750
      GOTO (38,36,34),I3                                                RTSP 760
   34 TR=TR-1.570796                                                    RTSP 770
   36 TR=TR-3.141593                                                    RTSP 780
      IF(TR.LT.0.0)TR=TR+6.283185                                       RTSP 790
      PL=1.570796-PL                                                    RTSP 800
C                           ADJUST OUTPUT TO REQUIRED UNITS             RTSP 810
   38 IF(J2)40,40,50                                                    RTSP 820
   40 T(K2)=TR                                                          RTSP 830
      P(K2)=PL                                                          RTSP 840
      GOTO 100                                                          RTSP 850
   50 T(K2)=TR*RTD                                                      RTSP 860
      P(K2)=PL*RTD                                                      RTSP 870
  100 CONTINUE                                                          RTSP 880
      RETURN                                                            RTSP 890
      END                                                               RTSP 900
```

```
C
C ......................................................................
C
C
C SUBROUTINE RTXYEQ
C
C PURPOSE
C     TO CONVERT A DIRECTION GIVEN AS A POINT IN 3-DIMENSIONAL SPACE
C     TO COORDINATES ON AN EQUAL-AREA PROJECTION.
C
C AUTHOR
C     JOHN RAMSDEN, DEPARTMENT OF GEOLOGY, UNIVERSITY OF ALBERTA,
C     ALBERTA, CANADA.
C
C DATE
C     1 OCTOBER 1974
C
C USAGE
C     CALL RTXYEQ(X,Y,Z,J1,RA,XP,YP,J2,N,IER)
C
C DESCRIPTION OF PARAMETERS
C     X, Y, Z  VECTORS OF COORDINATES IN 3-SPACE.
C     J1       SPACING OF INPUT WORDS IN X, Y, AND Z.
C     RA       RADIUS OF PROJECTION.
C     XP, YP   VECTORS OF COORDINATES OF THE POINT ON THE PROJECTION.
C     J2       SPACING OF OUTPUT WORDS IN XP, YP.
C     N        NUMBER OF POINTS GIVEN.
C     IER      ERROR CODE: 0 NO ERROR.
C                          1 AT LEAST ONE INPUT POINT WAS CLOSER TO
C                            THE ORIGIN THAN 1.0E-6. XP AND YP FOR
C                            THE POINT WERE SET TO 0.0.
C                          2 EITHER J1 OR J2 WAS 0.
C
C METHOD
C     THE PROJECTION IS LAMBERT'S EQUAL-AREA PROJECTION AND IS
C     PARALLEL TO THE X-Y PLANE.  THE XP AXIS PARALLELS THE Y AXIS;
C     THE YP AXIS PARALLELS THE X AXIS.  THE POINT PROJECTED IS THE
C     POINT OF INTERSECTION OF THE LINE FROM THE ORIGIN TO THE POINT
C     (X,Y,Z) WITH THE SPHERE OF RADIUS RA CENTRED AT THE ORIGIN.
C     FOR EACH POINT THE PROGRAM FIRST COMPUTES THE DISTANCE FROM THE
C     ORIGIN TO THE POINT, THEN DIVIDES THE COORDINATES BY THIS
C     DISTANCE TO OBTAIN DIRECTION COSINES A, B, G. X AND Y ARE THEN
C     COMPUTED USING THE EQUATIONS
C                X=BT   :       Y=AT
C     WHERE T=R*SQRT((1-G)/(1-G*G)) AND R=RADIUS OF PROJECTION.
C
C REMARKS
C     PROVIDED J2 IS NOT GREATER THAN J1, XP AND YP MAY OCCUPY THE
C     SAME LOCATIONS AS ANY TWO OF X, Y AND Z.   OTHERWISE THE INPUT
C     VECTORS ARE NOT DESTROYED.
C     MINIMUM SIZE FOR THE VECTORS XP AND YP IS J2*N.
C
C SUBPROGRAMS REQUIRED
C     NONE.
C
C ......................................................................
C
C
      SUBROUTINE RTXYEQ(X,Y,Z,J1,RA,XP,YP,J2,N,IER)
      REAL X(1),Y(1),Z(1),XP(1),YP(1)                                   RTXE 600
C                                                                       RTXE 610
      IER=0                                                             RTXE 620
      IF(J1.EQ.0.OR.J2.EQ.0)GOTO200                                     RTXE 630
      I1=1-J1                                                           RTXE 640
      I2=1-J2                                                           RTXE 650
      DO 100 I=1,N                                                      RTXE 660
      I1=I1+J1                                                          RTXE 670
      I2=I2+J2                                                          RTXE 680
      XI=X(I1)                                                          RTXE 690
      YI=Y(I1)                                                          RTXE 700
      ZI=Z(I1)                                                          RTXE 710
      R=SQRT(XI*XI+YI*YI+ZI*ZI)                                         RTXE 720
      IF(R.LT.1.0E-6)GOTO50                                             RTXE 730
      XI=XI/R                                                           RTXE 740
      YI=YI/R                                                           RTXE 750
      ZI=ZI/R                                                           RTXE 760
      T=RA*SQRT((1.0-ZI)/(1.0-ZI*ZI))                                   RTXE 770
      XP(I2)=YI*T                                                       RTXE 780
      YP(I2)=XI*T                                                       RTXE 790
      GOTO100                                                           RTXE 800
   50 IER=1                                                             RTXE 810
      XP(I2)=0.0                                                        RTXE 820
      YP(I2)=0.0                                                        RTXE 830
  100 CONTINUE                                                          RTXE 840
      RETURN                                                            RTXE 850
  200 IER=2                                                             RTXE 860
      RETURN                                                            RTXE 870
      END                                                               RTXE 880
```

```
C
C ...............................................................
C
C SUBROUTINE:  SCOM1
C
C PURPOSE
C     TO COMPUTE THE SUMS OF THE COMPONENTS OF A SET OF VECTORS
C
C AUTHOR
C     JOHN RAMSDEN, DEPARTMENT OF GEOLOGY, UNIVERSITY OF ALBERTA,
C     EDMONTON, ALBERTA, CANADA.
C
C DATE
C     1 JULY 1974
C
C USAGE:
C      CALL SCOM1(DC,W,N,V,JM,U)
C
C PARAMETERS:
C     DC 3*N MATRIX OF DIRECTION COSINES
C     W  VECTOR OF WEIGHTS
C     N  NUMBER OF DATA POINTS
C     V  VECTOR OF LENGTH 2 OR 3 - SEE JM
C     JM =0 IF DIRECTIONS USED AS GIVEN
C        =1 IF DIRECTIONS TO BE ADJUSTED AND V CONTAINS TREND AND PLUNGE
C           OF CENTRE OF HEMISPHERE IN DEGREES
C        =2 IF DIRECTIONS TO BE ADJUSTED AND V CONTAINS TREND AND PLUNGE
C           OF CENTRE OF HEMISPHERE IN RADIANS
C        =3 IF DIRECTIONS TO BE ADJUSTED AND V CONTAINS DIRECTION
C           COSINES OF CENTRE OF HEMISPHERE
C     U  VECTOR OF LENGTH 3 IN WHICH SUMS OF COMPONENTS ARE PLACED.
C
C SUBROUTINES CALLED:
C     SUBROUTINE "ANDC" FROM JR SUBROUTINE LIBRARY.
C
C ...............................................................
C
      SUBROUTINE SCOM1(DC,W,N,V,JM,U)
      LOGICAL*1 ASIS
      REAL DC(3,1),W(1),U(3),V(3)
C
      IF(JM) 100,100,50
   50 ASIS=.FALSE.
      GOTO(60,70,80),JM
   60 CALL ANDC(V,V(2),1,1,AC,BC,GC,1,1)
      GOTO150
   70 CALL ANDC(V(1),V(2),-1,1,AC,BC,GC,1,1)
      GOTO150
   80 AC=V(1)
      BC=V(2)
      GC=V(3)
      GOTO150
  100 ASIS=.TRUE.
  150 CONTINUE
C
      RX=0.0
      RY=0.0
      RZ=0.0
      DO 210 J=1,N
      X=DC(1,J)                                                         SCOM 600
      Y=DC(2,J)                                                         SCOM 610
      Z=DC(3,J)                                                         SCOM 620
      WJ=W(J)                                                           SCOM 630
      IF(ASIS)GOTO200                                                   SCOM 640
      C=X*AC+Y*BC+Z*GC                                                  SCOM 650
      IF(C)170,200,200                                                  SCOM 660
  170 X=-X                                                              SCOM 670
      Y=-Y                                                              SCOM 680
      Z=-Z                                                              SCOM 690
  200 RX=RX+WJ*X                                                        SCOM 700
      RY=RY+WJ*Y                                                        SCOM 710
      RZ=RZ+WJ*Z                                                        SCOM 720
  210 CONTINUE                                                          SCOM 730
      U(1)=RX                                                           SCOM 740
      U(2)=RY                                                           SCOM 750
      U(3)=RZ                                                           SCOM 760
      RETURN                                                            SCOM 770
C     DEBUG SUBTRACE,SUBCHK                                             SCOM 780
      END                                                               SCOM 790
```

```
C
C ...............................................................................
C
C SUBROUTINE SRF
C
C PURPOSE
C    TO CONVERT REAL NUMBERS TO INTEGERS, MULTIPLYING BY A GIVEN
C    POWER OF 10 AND ROUNDING.
C
C USAGE
C    CALL SRF(RL,IN,N,IF,MAX)
C
C DESCRIPTION OF PARAMETERS
C
C    RL      VECTOR OF GIVEN REAL NUMBERS
C    IN      VECTOR OF RETURNED INTEGERS
C    N       NUMBER OF NUMBERS TO BE PROCESSED
C    IF      POWER OF 10 BY WHICH NUMBERS ARE TO BE MULTIPLIED
C    MAX     MAXIMUM VALUE TO WHICH RESULTING INTEGERS ARE
C            TRUNCATED
C
C REMARKS
C    RL AND IN MAY BE IN THE SAME LOCATION
C    GIVEN NUMBERS MAY BE POSITIVE OR NEGATIVE
C    RETURNED INTEGERS ARE IN THE RANGE -MAX TO MAX INCLUSIVE
C
C SUBROUTINES REQUIRED
C    NONE
C
C METHOD
C    EACH REAL NUMBER IS MULTIPLIED BY 10**IF AND THEN ROUNDED.
C    IF THE RESULTING INTEGER IS > MAX, IT IS SET TO MAX.  IF IT
C    IS < -MAX IT IS SET TO -MAX.
C
C ...............................................................................
C
      SUBROUTINE SRF(RL,IN,N,IF,MAX)
      REAL RL(N)
      INTEGER IN(N)
C
      F=10.0**IF
      DO 200 J=1,N
      INJ=RL(J)*F+SIGN(0.5,RL(J))
      IF(INJ.GT.MAX) INJ=MAX
      IF(INJ.LT.-MAX) INJ=-MAX
      IN(J)=INJ
  200 CONTINUE
      RETURN
      END
```

```
C
C ..................................................................
C
C SUBROUTINE SYMM
C
C PURPOSE
C     TO USE BINGHAM'S TESTS BASED ON THE EIGENVALUES OF THE DATA CROSS
C     PRODUCT MATRIX TO PERFORM AN ANALYSIS OF THE SYMMETRY OF A SET
C     OF LINES.
C
C AUTHOR
C     JOHN RAMSDEN, DEPARTMENT OF GEOLOGY, UNIVERSITY OF ALBERTA,
C     EDMONTON, ALBERTA, CANADA.
C
C DATE
C     1 JULY 1974
C
C USAGE
C     CALL SYMM(U1,U2,U3,N,IR,K)
C
C PARAMETERS
C     U1,U2,U3  EIGENVALUES SCALED SO THAT U1+U2+U3=1.
C     N         NUMBER OF DATA POINTS
C     IR        TEST RESULT:
C               0 HYPOTHESIS K1=K2=K3 ACCEPTED
C               1 HYPOTHESIS K2=K3    ACCEPTED
C               2 HYPOTHESIS K1=K2    ACCEPTED
C               3 ALL HYPOTHESES REJECTED
C     K         PRINTOUT PARAMETER:
C               K<0  NO PRINTOUT
C               K>=0 RESULTS PRINTED ON I/O UNIT K.
C
C METHOD
C     TESTS ARE TAKEN FROM BINGHAM, C., 1964, DISTRIBUTIONS ON THE
C     SPHERE AND ON THE PROJECTIVE PLANE, PH.D. THESIS, YALE UNIVERSITY.
C     IT IS POSSIBLE THAT K1=K2=K3 IS REJECTED WHILE NEITHER K2=K3 NOR
C     K1=K2 IS REJECTED.  IN THIS CASE THE PROGRAM CHOOSES TO ACCEPT
C     WHICHEVER OF THE LAST TWO HYPOTHESES YIELDS THE SMALLER STATISTIC.
C
C SUBPROGRAMS AND FUNCTION SUBPROGRAMS REQUIRED
C     NONE.
C
C ..................................................................
C
      SUBROUTINE SYMM(U1,U2,U3,N,IR,K)
    1 FORMAT(     ' '/'TEST OF HYPOTHESIS K1=K2=K3: TEST STATISTIC='
     *,F10.4,'      5% POINT=11.0705')
    4 FORMAT(         ' '/1X,'TEST OF HYPOTHESIS K2=K3',        ': TEST STA
     *TISTIC=',F10.4,'      5% POINT=5.99147')
    5 FORMAT(         ' '/1X,'TEST OF HYPOTHESIS K1=K2',        ': TEST STA
     *TISTIC=',F10.4,'      5% POINT=5.99147')
    7 FORMAT(' '/' HYPOTHESIS K1=K2=K3 ACCEPTED')
    8 FORMAT(' '/' HYPOTHESIS K2=K3 ACCEPTED')
    9 FORMAT(' '/' HYPOTHESIS K1=K2 ACCEPTED')
   10 FORMAT(' '/' ALL HYPOTHESES REJECTED')
   11 FORMAT(' ')
C
C
      IR=0
      I1=1
      I2=1
      I3=1
      IF(K.GE.0)WRITE(K,11)
C
C          TEST FOR HYPOTHESIS K1=K2=K3 ("CUBIC" SYMMETRY)
C
      TEST=((ALOG(3.0*U1))**2+(ALOG(3.*U2))**2+(ALOG(3.*U3))**2)*N*5./6.
      IF(K.GE.0)WRITE(K,1)TEST
      IF(TEST-11.0705) 150,140,140
  140 I1=2
C
C          TEST HYPOTHESIS K2=K3 ("TETRAGONAL" SYMMETRY(1))
C
  150 TEST=ALOG(U2/U3)
      TEST=TEST*TEST*N/3.0
      IF(K.GE.0)WRITE(K,4)TEST
      IF(TEST-5.99147) 160,180,180
  160 R1=TEST
      GOTO190
  180 I2=2
C
C          TEST HYPOTHESIS K1=K2 ("TETRAGONAL" SYMMETRY(2))
C
  190 TEST=ALOG(U1/U2)
      TEST=TEST*TEST*N/3.0
      IF(K.GE.0)WRITE(K,5)TEST
      IF(TEST-5.99147) 300,220,220
  220 I3=2
C
C-----ANALYZE TEST RESULTS
C
  300 IF(I1.EQ.1)GOTO400
      GOTO(320,340),I2
  320 GOTO(500,520),I3
  340 GOTO(540,600),I3
C
C-----SET RESULT CODE AND WRITE MESSAGE
C
  400 IR=0
      IF(K.GE.0)WRITE(K,7)
      RETURN
  500 IF(R1-TEST)520,520,540
  520 IR=1
      IF(K.GE.0)WRITE(K,8)
      RETURN
  540 IR=2
      IF(K.GE.0)WRITE(K,9)
      RETURN
  600 IR=3
      IF(K.GE.0)WRITE(K,10)
      RETURN
C     DEBUG SUBCHK,SUBTRACE
      END
```

```
C
C ......................................................................
C
C FUNCTION TAB1
C
C PURPOSE
C     TO LOOK UP AND INTERPOLATE A ONE-DIMENSIONAL TABLE.
C
C AUTHOR
C     JOHN RAMSDEN, DEPARTMENT OF GEOLOGY, UNIVERSITY OF ALBERTA,
C     EDMONTON, ALBERTA, CANADA.
C
C DATE
C     1 JULY 1974
C
C USAGE
C     Y=TAB1(X,A,B,N)
C
C PARAMETERS
C     X    GIVEN VALUE WITH WHICH TO ENTER THE TABLE
C     A    VECTOR CONTAINING THE COLUMN OF THE TABLE IN WHICH X IS TO
C          BE FOUND.
C     B    VECTOR CONTAINING THE COLUMN OF THE TABLE FROM WHICH RETURNED
C          VALUE IS TO BE OBTAINED.
C     N    NUMBER OF ROWS (ENTRIES) IN THE TABLE.
C     TAB1 THE VALUE OF THE FUNCTION IS THE VALUE OBTAINED FROM THE
C          TABLE.
C
C REMARKS
C     IF X IS BEYOND THE RANGE OF THE TABLE THE RETURNED VALUE IS THE
C     EXTREME VALUE OF B IN THE TABLE.
C
C SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C     NONE
C
C
C ......................................................................
C
      FUNCTION TAB1(X,A,B,N)
      REAL A(N),B(N)
      IF(X.LE.A(1))GOTO200
      IF(X.GE.A(N))GOTO300
      J=1
  100 IF(X-A(J))120,140,160
  120 TAB1=B(J-1)+(B(J)-B(J-1))*(X-A(J-1))/(A(J)-A(J-1))
      RETURN
  140 TAB1=B(J)
      RETURN
  160 J=J+1
      GOTO100
  200 TAB1=B(1)
      RETURN
  300 TAB1=B(N)
      RETURN
      END
```

```
C
C ...............................................................
C
C SUBROUTINE TPXYEQ
C
C PURPOSE
C      TO CONVERT A DIRECTION GIVEN AS TREND AND PLUNGE TO COORDINATES
C      ON AN EQUAL-AREA PROJECTION.
C
C AUTHOR
C      JOHN RAMSDEN, DEPARTMENT OF GEOLOGY, UNIVERSITY OF ALBERTA,
C      EDMONTON, ALBERTA, CANADA.
C
C DATE
C      1 OCTOBER 1974
C
C USAGE
C      CALL TPXYEQ(T,P,J1,RA,XP,YP,J2,N,IER)
C
C DESCRIPTION OF PARAMETERS
C      T, P      VECTORS OF TRENDS AND PLUNGES.
C      J1        INPUT PARAMETER: |J1|=SPACING OF INPUT WORDS IN T AND P.
C                                  J1<0 IF INPUT UNITS ARE DEGREES.
C                                  J1>0 IF INPUT UNITS ARE RADIANS.
C      RA        RADIUS OF PROJECTION.
C      XP, YP    VECTORS OF COORDINATES ON PPOJECTION.
C      J2        SPACING OF OUTPUT WOPDS IN XP, YP.
C      N         NUMBER OF POINTS GIVEN.
C      IER       ERROR CODE: 0 NO ERROR.
C                            2 EITHER J1 OR J2 WAS ZERO.
C
C METHOD
C      THE PROJECTION IS LAMBERT'S EQUAL-AREA PROJECTION AND IS
C      HORIZONTAL.  THE XP AXIS IS EAST; THE YP AXIS IS NORTH.  TREND
C      IS MEASURED CLOCKWISE FROM NORTH.
C
C REMARKS
C      XP AND YP MAY OCCUPY THE SAME LOCATIONS AS T AND P PROVIDED
C      J2 IS NOT GREATER THAN J1.  OTHERWISE THE INPUT VECTORS ARE NOT
C      DESTROYED.
C      MINIMUM SIZE FOR THE OUTPUT VECTORS XP AND YP IS J2*N.
C
C SUBPROGRAMS REQUIRED
C      NONE.
C
C ...............................................................
C
C
      SUBROUTINE TPXYEQ(T,P,J1,RA,XP,YP,J2,N,IER)
      LOGICAL*1 RADIAN
      REAL T(1),P(1),XP(1),YP(1)
C
      RADIAN=.TRUE.
      IER=0
      IF(J1.EQ.0.OR.J2.EQ.0)GOTO200
      JJ1=J1
      IF(JJ1.GT.0)GOTO30
      RADIAN=.FALSE.
      JJ1=-JJ1
```

```
   30 I1=1-JJ1                                                          TPXE 600
      I2=1-J2                                                           TPXE 610
      DO 100 I=1,N                                                      TPXE 620
      I1=I1+JJ1                                                         TPXE 630
      I2=I2+J2                                                          TPXE 640
      TI=T(I1)                                                          TPXE 650
      PI=P(I1)                                                          TPXE 660
      IF(RADIAN)GOTO50                                                  TPXE 670
      TI=TI*1.745329E-2                                                 TPXE 680
      PI=PI*1.745329E-2                                                 TPXE 690
   50 S=RA*SQRT(1.0-SIN(PI))                                            TPXE 700
      XP(I2)=S*SIN(TI)                                                  TPXE 710
      YP(I2)=S*COS(TI)                                                  TPXE 720
  100 CONTINUE                                                          TPXE 730
      RETURN                                                            TPXE 740
  200 IER=2                                                             TPXE 750
      RETURN                                                            TPXE 760
      END                                                               TPXE 770
```

```
C
C ...........................................................................
C
C SUBROUTINE VRM1
C
C PURPOSE
C     TO RETURN THE ROTATION MATRIX THAT WILL ROTATE A GIVEN DIRECTION
C     TO THE VERTICAL.
C
C
C AUTHOR
C     JOHN RAMSDEN, DEPARTMENT OF GEOLOGY, UNIVERSITY OF ALBERTA,
C     EDMONTON, ALBERTA, CANADA.
C
C DATE
C     1 JULY 1974
C
C USAGE
C     CALL VRM1(V,M,J)
C
C PARAMETERS
C     V VECTOR OF LENGTH 2 OR 3 - SEE J.
C     M RETURNED ROTATION MATRIX.
C     J =1 IF V(1),V(2) CONTAIN TREND AND PLUNGE IN DEGREES
C       =2 IF V(1),V(2) CONTAIN TREND AND PLUNGE IN RADIANS
C       =3 IF V(1),V(2),V(3) CONTAIN DIRECTION COSINES
C
C METHOD
C     THE TREND AND PLUNGE OF THE AXIS OF ROTATION AND THE ANGLE OF
C     ROTATION ARE COMPUTED, AND THE SUBROUTINE "ROTMAT" IS CALLED.
C
C SUBROUTINES REQUIRED
C     SUBROUTINES "RTSP" AND "ROTMAT" FROM JR SUBROUTINE LIBRARY.
C
C
C ...........................................................................
C
      SUBROUTINE VRM1(V,M,J)
      REAL V(3),M(9)
C
      GOTO(100,200,300),J
  100 TR=V(1)*1.745329E-2
      PL=V(2)*1.745329E-2
      GOTO400
  200 TR=V(1)
      PL=V(2)
      GOTO400
  300 CALL RTSP(V(1),V(2),V(3),1,TR,PL,R,-1,-1,1)
  400 TAX=TR+1.570796
      PAX=0.0
      AR=PL-1.570796
      CALL ROTMAT(TAX,PAX,AR,M)
      RETURN
      DEBUG SUBTRACE,SUBCHK
      END
```

```
C
C .......................................................................
C
C SUBROUTINE WRITEK
C
C PURPOSE
C     TO WRITE A SET OF HALFWORD INTEGERS IN A SPECIAL FORMAT
C
C AUTHOR
C     JOHN RAMSDEN   DEPT OF GEOLOGY   UNIVERSITY OF ALBERTA
C
C DATE WRITTEN
C     MARCH 1974
C
C USAGE
C     CALL WRITEK(IOU,TITLE,LENFIL,LENKEY,KEY)
C
C DESCRIPTION OF PARAMETERS
C     IOU    I/O UNIT TO BE WRITTEN ON
C     TITLE  120-BYTE TITLE
C     LENFIL SIZE OF UNIVERSE TO WHICH THE SET BELONGS
C     LENKEY SIZE OF SET
C     KEY    SET
C
C FORMAT OF OUTPUT
C     ALL OUTPUT RECORDS ARE 80 BYTES LONG AND HAVE A SEQUENCE NUMBER IN
C     COLUMNS 78-80.  THE FIRST RECORD CONTAINS THE FIRST 72 BYTES OF
C     'TITLE' IN COLUMNS 1-72.  THE SECOND RECORD CONTAINS THE LAST 48
C     BYTES OF 'TITLE' IN COLUMNS 1-48, 'LENFIL' IN COLUMNS 53-56, AND
C     'LENKEY' IN COLUMNS 61-64.  ALL SUBSEQUENT RECORDS CONTAIN 18
C     NUMBERS FROM THE SET IN COLUMNS 1-72.
C
C SUBROUTINES REQUIRED
C     NONE
C
C .......................................................................
C
      SUBROUTINE WRITEK(IOU,TITLE,LENFIL,LENKEY,KEY)
C
    1 FORMAT(18A4,7X,'1')
    2 FORMAT(12A4,4X,I4,4X,I4,15X,'2')
    3 FORMAT(18I4,5X,I3)
C
      INTEGER TITLE(30)
      INTEGER*2 KEY(5018)
C
      WRITE(IOU,1) (TITLE(I),I=1,18)
      WRITE(IOU,2) (TITLE(I),I=19,30),LENFIL,LENKEY
      IF(LENKEY.EQ.0) RETURN
C
      KA=LENKEY+1
      KB=LENKEY+18
      DO 100 I=KA,KB
  100 KEY(I)=0
C
      KA=1
      KB=18
      N=3
  200 WRITE(IOU,3) (KEY(I),I=KA,KB),N
      KA=KA+18
      IF(KA.GT.LENKEY) RETURN
      KB=KB+18
      N=N+1
      GOTO 200
C
      END
```

```
C
C .......................................................................
C
C SUBROUTINE XKF
C
C PURPOSE
C      TO ESTIMATE THE CONCENTRATION PARAMETER FOR FISHER'S DISTRIBUTION
C      GIVEN R=Q/S WHERE Q IS THE LENGTH OF THE RESULTANT OF A SET OF
C      VECTORS AND S IS THE SUM OF THE LENGTHS OF THE VECTORS.
C
C AUTHOR
C      JOHN RAMSDEN, DEPARTMENT OF GEOLOGY, UNIVERSITY OF ALBERTA,
C      EDMONTON, ALBERTA, CANADA.
C
C DATE
C      1 JULY 1974
C
C USAGE
C      CALL XKF(R,N,AK,IE)
C
C DESCRIPTION OF PARAMETERS
C      R       PROPORTIONAL LENGTH OF RESULTANT =Q/S
C      N       NUMBER OF VECTORS IN SET
C      AK      ESTIMATED CONCENTRATION PARAMETER
C      IE      ERROR CODE:
C              0 NO ERROR
C              1 R OUTSIDE RANGE 0 TO 1.  AK SET TO 0.0.
C
C METHOD
C      THE TABLE GIVEN BY MARDIA (1972,P.322) IS USED TO OBTAIN AK IF
C      R<0.9.  OTHERWISE THE ESTIMATE (1-2/N)/(1-R) IS USED.
C
C SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C      FUNCTION "TAB1" FROM JR SUBROUTINE LIBRARY.
C
C REFERENCE
C      MARDIA, K. V. 1972. STATISTICS OF DIRECTIONAL DATA. LONDON & NEW
C      YORK, ACADEMIC PRESS. 357PP.
C
C
C .......................................................................
C
      SUBROUTINE XKF(R,N,AK,IE)
      REAL TABR(90)/.00,.01,.02,.03,.04,.05,.06,.07,.08,.09,.10,.11,.12,
     *.13,.14,.15,.16,.17,.18,.19,.20,.21,.22,.23,.24,.25,.26,.27,.28,
     *.29,.30,.31,.32,.33,.34,.35,.36,.37,.38,.39,.40,.41,.42,.43,.44,
     *.45,.46,.47,.48,.49,.50,.51,.52,.53,.54,.55,.56,.57,.58,.59,.60,
     *.61,.62,.63,.64,.65,.66,.67,.68,.69,.70,.71,.72,.73,.74,.75,.76,
     *.77,.78,.79,.80,.81,.82,.83,.84,.85,.86,.87,.88,.89/
      REAL TABK(90)/0.0,0.03,.06001,.09005,.12012,.15023,.18039,.21062,
     *.24093,.27132,.30182,.33242,.36315,.39402,.42503,.45621,.48756,
     *.51909,.55083,.58278,.61497,.64740,.68009,.71306,.74632,.77990,
     *.81381,.84806,.88269,.91771,.95315,.98902,1.02536,1.06218,1.09951,
     *1.13739,1.17584,1.21490,1.25459,1.29497,1.33605,1.37789,1.42053,
     *1.46401,1.50839,1.55372,1.60005,1.64745,1.69599,1.74573,1.79676,
     *1.84915,1.90300,1.95842,2.01550,2.07437,2.13515,2.19799,2.26304,
     *2.33049,2.40050,2.47331,2.54914,2.62825,2.71093,2.79751,2.88836,
     *2.98389,3.08456,3.19091,3.30354,3.42314,3.55051,3.68655,3.83232,
     *3.98905,4.15819,4.34143,4.54076,4.75857,4.99772,5.26167,5.55463,
     *5.88181,6.24971,6.66652,7.14279,7.69228,8.33333,9.09091/          XKF  600
C                                                                       XKF  610
C                                                                       XKF  620
      IF(P) 10,20,30                                                    XKF  630
   10 IF=1                                                              XKF  840
   20 AK=0.0                                                            XKF  650
      GOTO900                                                           XKF  660
   30 IF(P-0.9)40,50,50                                                 XKF  670
   40 AK=TAB1(P,TABR,TABK,90)                                           XKF  680
      GOTO900                                                           XKF  690
   50 IF(P-1.0)60,70,10                                                 XKF  700
   60 AK=(1.0-2.0/N)/(1.0-R)                                            XKF  710
      GOTO900                                                           XKF  720
   70 AK=1.0E74                                                         XKF  730
  900 RETURN                                                            XKF  740
C     DEBUG SUBTRACE,SUBCHK                                             XKF  750
      END                                                               XKF  760
```

```
C
C ..........................................................................
C FUNCTION XK1
C
C PURPOSE
C      TO ESTIMATE THE CONCENTRATION PARAMETER FOR DIMROTH-WATSON'S
C      DISTRIBUTION FOR THE GIRDLE CASE GIVEN U1, THE SMALLEST EIGEN-
C      VALUE OF U=T/TR(T) WHERE TR(T) IS THE TRACE OF T AND T IS THE
C      DATA CROSS PRODUCT MATRIX.
C
C AUTHOR
C      JOHN RAMSDEN, DEPARTMENT OF GEOLOGY, UNIVERSITY OF ALBERTA,
C      EDMONTON, ALBERTA, CANADA.
C
C DATE
C      1 JULY 1974
C
C USAGE
C      XK=XK1(X)
C
C PARAMETERS
C      X   THE SMALLEST EIGENVALUE OF U=T/TR(T)
C      XK1 THE VALUE OF THE FUNCTION IS THE ESTIMATED CONCENTRATION
C          PARAMETER.
C
C METHOD
C      TABLE LOOKUP IF 0.05<X<0.32; TABLE FROM MARDIA, 1972, P.323.
C      OTHERWISE FORMULAS GIVEN BY MARDIA (1972, P.254) ARE USED.
C
C SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C      FUNCTION "TAB1" FROM JR SUBROUTINE LIBRARY.
C
C REFERENCE
C      MARDIA, K. V., 1972, STATISTICS OF DIRECTIONAL DATA. LONDON AND
C      NEW YORK, ACADEMIC PRESS. 357PP.
C
C
C
C ..........................................................................
C
      FUNCTION XK1(X)
      REAL U(55)/.05,.055,.06,.065,.07,.075,.08,.085,.09,.095,.1,.105,
     $.11,.115,.12,.125,.13,.135,.14,.145,.15,.155,.16,.165,.17,.175,
     *.18,.185,.19,.195,.2,.205,.21,.215,.22,.225,.23,.235,.24,.245,.25,
     *.255,.26,.265,.27,.275,.28,.285,.29,.295,.3,.305,.31,.315,.32/
      REAL K(55)/9.99,9.09,8.33,7.68,7.13,6.64,6.22,5.84,5.50,5.19,4.91,
     *4.65,4.41,4.20,3.99,3.80,3.62,3.46,3.30,3.15,3.01,2.87,2.74,2.62,
     *2.50,2.39,2.28,2.17,2.07,1.97,1.87,1.78,1.69,1.61,1.52,1.44,1.36,
     *1.28,1.20,1.13,1.05,.98,.91,.84,.77,.71,.64,.58,.51,.45,.39,.33,
     *.27,.21,.15/
C
      IF(X.GT.0.320)GOTO200
      IF(X.LT.0.050)GOTO300
C
      XK1=-TAB1(X,U,K,49)
      RETURN
C
  200 XK1=-(15.0-45.0*X)/4.0
      RETURN
C
  300 XK1=-1.0/(2.0*X)
      RETURN
C
C     DEBUG SUBTRACE,SUBCHK
      END
```

```
C
C ......................................................................
C
C FUNCTION XK3
C
C PURPOSE
C      TO ESTIMATE THE CONCENTRATION PARAMETER OF DIMROTH-WATSON'S
C      DISTRIBUTION FOR THE BIPOLAR CASE GIVEN U3, THE LARGEST EIGEN-
C      VALUE OF THE MATRIX U=T/TR(T) WHERE TR(T) IS THE TRACE OF T AND
C      T IS THE CROSS PRODUCT MATRIX REPRESENTING THE SET OF AXES.
C
C AUTHOR
C      JOHN RAMSDEN, DEPARTMENT OF GEOLOGY, UNIVERSITY OF ALBERTA,
C      EDMONTON, ALBERTA, CANADA.
C
C DATE
C      1 JULY 1974
C
C USAGE
C      XK=XK3(X)
C
C PARAMETERS
C      X   THE LARGEST EIGENVALUE OF U=T/TR(T)
C      XK3 THE VALUE OF THE FUNCTION IS THE COMPUTED CONCENTRATION
C          PARAMETER
C
C METHOD
C      TABLE LOOKUP IF 0.351<X<0.948;  TABLE FROM MARDIA (1972, P.324).
C      OTHERWISE BY FORMULAS GIVEN BY MARDIA (1972, P.254).
C
C SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C      NONE
C
C REFERENCE
C      MARDIA, K. V. 1972. STATISTICS OF DIRECTIONAL DATA. LONDON AND NEW
C      YORK, ACADEMIC PRESS. 357PP.
C
C
C ......................................................................
C
      FUNCTION XK3(X)
      REAL U(45)/.351,.370,.389,.409,.429,.450,.470,.491,.511,.531,.551,
     *.571,.590,.608,.626,.643,.660,.676,.690,.705,.718,.731,.743,.754,
     *.764,.788,.806,.825,.839,.851,.862,.871,.879,.886,.892,.903,.912,
     *.919,.925,.930,.935,.939,.942,.946,.948/
      REAL K(45)/.2,.4,.6,.8,1.,1.2,1.4,1.6,1.8,2.0,2.4,2.6,2.8,3.0,3.2,
     *3.4,3.6,3.8,4.0,4.2,4.4,4.6,4.8,5.0,5.5,6.0,6.5,7.0,7.5,8.0,8.5,
     *9.0,9.5,10.0,11.,12.,13.,14.,15.,16.,17.,18.,19.,20./
C
      IF(X.LT.0.351)GOTO200
      IF(X.GT.0.948)GOTO300
C
C
   99 J=1
  100 IF(X-U(J))120,140,160
  120 XK3=K(J-1)+(K(J)-K(J-1))*(X-U(J-1))/(U(J)-U(J-1))
      RETURN
  140 XK3=K(J)
      RETURN
  160 J=J+1
      GOTO100
  200 XK3=(45.0*X-15.0)/4.0
      RETURN
  300 XK3=1.0/(1.0-X)
      RETURN
C     DEBUG SUBTRACE,SUBCHK
      END
```

Documentation of System Subroutines

Name	PLOTS
Usage	CALL PLOTS
Action	Initializes plot routines

Name	PLOT
Usage	CALL PLOT(X,Y,J)
Action	The pen is moved from its present position to the point X,Y. J=2 pen down J=3 pen up J<0 redefine plot origin as X,Y. J=999 terminate plot.

Name	SYMBOL
Usage	CALL SYMBOL(X,Y,H,A,T,N)
Action	The N characters stored in the first N bytes of the array A are drawn with a height H, starting at the point X,Y so that they make an angle T degrees with the positive X-axis.

Name	NUMBER
Usage	CALL NUMBER(X,Y,H,R,T,J)
Action	The real number R is drawn with height H at the point X,Y so that it makes an angle T degrees with the positive X-axis. J is the number of decimal places required.

Name	POLAR
Usage	CALL POLAR(R,THETA,N,K,J,RAD,SCALE)
Action	A series of N points whose polar coordinates are stored in every Kth location, starting with the first, of the vectors R and THETA are plotted as points if J<0 or connected by a line if J>0. If RAD=0.0, SCALE is the number of data points per inch. If RAD>0.0, the plot is scaled so that the maximum radius of the plot is <=RAD.

Name	URAND
Usage	R=URAND(Z)
Action	The value of the function is a random number on (0,1). Z is initially zero.

Name	WHERE
Usage	CALL WHERE(X,Y,F)
Action	The current pen position is returned in X,Y; the current multiplying factor is returned in F.

Appendix 3

Table of Density Estimator Data

```
kρ=50.0 n=50
 DISTANCE              0.0        5.0000    10.0000    15.0000    20.0000    25.0000
 TRUE DENSITY         50.0000    41.3370    23.3924     9.1004     2.4514     0.4618
CONSTANT-AREA A/2π=2%
 MAXIMUM              39.0000    34.0000    28.0000    19.0000     9.0000     5.0000
 MINIMUM              23.0000    19.0000    12.0000     5.0000     2.0000     0.0
 MEAN                 30.8000    27.7200    20.0800    11.7200     5.1000     1.7800
 STD DEV               2.9966     3.4643     3.9530     3.0973     1.8434     1.1480
 MEAN/TRUE             0.6160     0.6706     0.8584     1.2879     2.0804     3.8545
 SD/MEAN               0.0973     0.1250     0.1969     0.2643     0.3615     0.6449
 LOG(MEAN/TRUE)       -0.2104    -0.1735    -0.0663     0.1099     0.3182     0.5860
 LOG(SD/MEAN)         -1.0119    -0.9032    -0.7058    -0.5779    -0.4420    -0.1905
CONSTANT-AREA A/2π=5%
 MAXIMUM              20.0000    19.2000    17.6000    14.4000    10.0000     4.8000
 MINIMUM              16.0000    14.8000    12.4000     8.0000     3.6000     1.2000
 MEAN                 18.4959    17.6879    15.0879    11.0719     6.9040     3.3120
 STD DEV               0.8068     0.8957     1.1265     1.4603     1.2723     0.8361
 MEAN/TRUE             0.3699     0.4269     0.6450     1.2166     2.8163     7.1719
 SD/MEAN               0.0436     0.0508     0.0747     0.1319     0.1843     0.2524
 LOG(MEAN/TRUE)       -0.4319    -0.3696    -0.1904     0.0852     0.4497     0.8556
 LOG(SD/MEAN)         -1.3603    -1.2945    -1.1269    -0.8798    -0.7345    -0.5978
CONSTANT-AREA A/2π=1%
 MAXIMUM              52.0000    50.0000    38.0000    22.0000    10.0000     4.0000
 MINIMUM              24.0000    16.0000    10.0000     4.0000     0.0        0.0
 MEAN                 39.2000    33.8000    22.0000    11.2000     3.9000     0.9000
 STD DEV               6.2000     8.0000     5.7000     4.2000     2.6000     1.0000
 MEAN/TRUE             0.7840     0.8177     0.9405     1.2307     1.5909     1.9489
 SD/MEAN               0.1582     0.2367     0.2591     0.3750     0.6667     1.1111
 LOG(MEAN/TRUE)       -0.1057    -0.0874    -0.0267     0.0902     0.2017     0.2898
 LOG(SD/MEAN)         -0.8009    -0.6258    -0.5865    -0.4260    -0.1761     0.0458
CONSTANT-AREA KAMB
 MAXIMUM               6.6000     6.6000     6.6000     6.6000     6.6000     5.9000
 MINIMUM               6.6000     6.4000     6.4000     6.0000     5.1000     4.2000
 MEAN                  6.5000     6.5000     6.5000     6.3000     5.9000     4.9000
 STD DEV               0.0        0.0        0.0        0.1000     0.3000     0.4000
 MEAN/TRUE             0.1300     0.1572     0.2779     0.6923     2.4068    10.6107
 SD/MEAN               0.0        0.0        0.0        0.0159     0.0508     0.0816
 LOG(MEAN/TRUE)       -0.8861    -0.8034    -0.5562    -0.1597     0.3814     1.0257
 LOG(SD/MEAN)        -10.0000   -10.0000   -10.0000    -1.7993    -1.2937    -1.0881
```

```
kp=50.0 n=50
 DISTANCE               0.0        5.0000     10.0000     15.0000     20.0000     25.0000
 TRUE DENSITY          50.0000    41.3370     23.3924      9.1004      2.4514      0.4618
CONSTANT-PROPORTION  Vo=1/7
 MAXIMUM               48.7344    42.2365     28.3027     17.5059     10.2159      6.2897
 MINIMUM               23.7961    19.5085     12.9071      8.6225      6.0147      3.9879
 MEAN                  34.2970    29.9247     19.7706     11.7478      7.3359      4.9845
 STD DEV                5.6820     5.5207      4.0025      1.7890      0.8770      0.5089
 MEAN/TRUE              0.6859     0.7239      0.8452      1.2909      2.9925     10.7936
 SD/MEAN                0.1657     0.1845      0.2024      0.1523      0.1195      0.1021
 LOG(MEAN/TRUE)        -0.1637    -0.1403     -0.0731      0.1109      0.4760      1.0332
 LOG(SD/MEAN)          -0.7808    -0.7340     -0.6937     -0.8173     -0.9225     -0.9910
CONSTANT-PROPORTION  Vo=1/10
 MAXIMUM               39.1884    32.9485     24.2390     14.7109      9.3088      6.3220
 MINIMUM               18.0700    16.0201     11.6072      8.5540      5.9802      4.4629
 MEAN                  28.4677    25.0931     16.8483     10.7388      7.2438      5.1890
 STD DEV                4.5365     3.9515      2.6002      1.2400      0.7097      0.4183
 MEAN/TRUE              0.5694     0.6070      0.7202      1.1800      2.9550     11.2365
 SD/MEAN                0.1594     0.1575      0.1543      0.1155      0.0980      0.0806
 LOG(MEAN/TRUE)        -0.2446    -0.2168     -0.1425      0.0719      0.4706      1.0506
 LOG(SD/MEAN)          -0.7976    -0.8028     -0.8115     -0.9375     -1.0089     -1.0936
CONSTANT-PROPORTION  Vo=1/20
 MAXIMUM               27.0000    23.2328     16.5194     11.3215      7.6889      5.5787
 MINIMUM               12.6346    11.5730      8.2574      6.3011      4.7125      3.6691
 MEAN                  20.2296    17.0085     12.0264      8.3798      6.1825      4.6946
 STD DEV                3.1428     2.4501      1.8070      1.0718      0.6639      0.4189
 MEAN/TRUE              0.4046     0.4115      0.5141      0.9208      2.5220     10.1659
 SD/MEAN                0.1554     0.1441      0.1503      0.1279      0.1074      0.0892
 LOG(MEAN/TRUE)        -0.3930    -0.3857     -0.2889     -0.0358      0.4017      1.0071
 LOG(SD/MEAN)          -0.8087    -0.8415     -0.8232     -0.8931     -0.9691     -1.0495
CONSTANT-PROPORTION  Vo=1/3
 MAXIMUM              131.3000    73.9000     42.0000     24.1000     11.8000      5.4000
 MINIMUM               21.5000    16.4000     16.2000      7.3000      4.1000      2.4000
 MEAN                  50.5000    40.0000     24.9000     12.4000      6.0000      3.4000
 STD DEV               20.0000    12.8000      8.4000      3.9000      1.4000      0.5000
 MEAN/TRUE              1.0100     0.9677      1.0644      1.3626      2.4476      7.3625
 SD/MEAN                0.3960     0.3200      0.3373      0.3145      0.2333      0.1471
 LOG(MEAN/TRUE)         0.0043    -0.0143      0.0271      0.1344      0.3887      0.8670
 LOG(SD/MEAN)          -0.4023    -0.4948     -0.4719     -0.5024     -0.6320     -0.8325
CONSTANT-PROPORTION  Vo=1/5
 MAXIMUM               65.7000    53.7000     37.0000     19.6000      9.8000      6.0000
 MINIMUM               27.6000    19.6000     14.0000      8.2000      5.1000      3.6000
 MEAN                  42.4000    35.0000     21.6000     11.6000      6.7000      4.3000
 STD DEV                8.8000     8.2000      5.1000      2.3000      0.9000      0.5000
 MEAN/TRUE              0.8480     0.8467      0.9234      1.2747      2.7331      9.3114
 SD/MEAN                0.2075     0.2343      0.2361      0.1983      0.1343      0.1163
 LOG(MEAN/TRUE)        -0.0716    -0.0723     -0.0346      0.1054      0.4367      0.9690
 LOG(SD/MEAN)          -0.6829    -0.6303     -0.6269     -0.7027     -0.8718     -0.9345
```

```
kp=50.0 n=50
 DISTANCE              0.0        5.0000     10.0000     15.0000     20.0000     25.0000
 TRUE DENSITY         50.0000    41.3370     23.3924      9.1004      2.4514      0.4618
FISHER-WEIGHTED Kw=242
 MAXIMUM              58.1000    48.9000     29.9000     21.0000     12.0000      4.1000
 MINIMUM              22.9000    15.9000     11.9000      2.8000      0.2000     0.0
 MEAN                 40.8000    34.4000     22.3000     10.9000      3.9000      0.9000
 STD DEV               8.0000     7.8000      6.7000      4.6000      2.6000      1.1000
 MEAN/TRUE             0.8160     0.8322      0.9533      1.1977      1.5909      1.9489
 SD/MEAN               0.1961     0.2267      0.3004      0.4220      0.6667      1.2222
 LOG(MEAN/TRUE)       -0.0883    -0.0798     -0.0208      0.0784      0.2017      0.2898
 LOG(SD/MEAN)         -0.7076    -0.6445     -0.5222     -0.3747     -0.1761      0.0872
FISHER-WEIGHTED Kw=100
 MAXIMUM              40.6000    36.9000     30.2000     17.2000     10.4000      4.1000
 MINIMUM              23.5000    18.6000     14.3000      6.3000      1.7000      0.3000
 MEAN                 33.1000    29.0000     20.1000     11.1000      4.8000      1.5000
 STD DEV               3.7000     4.3000      3.9000      2.7000      1.7000      0.8000
 MEAN/TRUE             0.6620     0.7016      0.8593      1.2197      1.9581      3.2482
 SD/MEAN               0.1118     0.1483      0.1940      0.2432      0.3542      0.5333
 LOG(MEAN/TRUE)       -0.1791    -0.1539     -0.0659      0.0863      0.2918      0.5116
 LOG(SD/MEAN)         -0.9516    -0.8289     -0.7121     -0.6140     -0.4508     -0.2730
FISHER-WEIGHTED Kw=50
 MAXIMUM              29.4000    27.1000     22.3000     15.1000      9.2000      4.8000
 MINIMUM              20.7000    17.7000     13.4000      8.0000      3.5000      1.2000
 MEAN                 24.9000    22.5000     17.1000     10.8000      5.6000      2.4000
 STD DEV               1.8000     2.2000      2.2000      1.7000      1.2000      0.7000
 MEAN/TRUE             0.4980     0.5443      0.7310      1.1868      2.2844      5.1971
 SD/MEAN               0.0723     0.0978      0.1287      0.1574      0.2143      0.2917
 LOG(MEAN/TRUE)       -0.3028    -0.2642     -0.1361      0.0744      0.3588      0.7158
 LOG(SD/MEAN)         -1.1409    -1.0098     -0.8906     -0.8030     -0.6690     -0.5351
FISHER-WEIGHTED Kw=25
 MAXIMUM              18.6000    17.6000     15.2000     11.7000      8.0000      5.1000
 MINIMUM              15.3000    13.9000     11.3000      7.9000      4.9000      2.6000
 MEAN                 16.6000    15.5000     12.9000      9.4000      6.1000      3.5000
 STD DEV               0.7000     0.9000      1.0000      0.9000      0.7000      0.5000
 MEAN/TRUE             0.3320     0.3750      0.5515      1.0329      2.4884      7.5790
 SD/MEAN               0.0422     0.0581      0.0775      0.0957      0.1148      0.1429
 LOG(MEAN/TRUE)       -0.4789    -0.4260     -0.2585      0.0141      0.3959      0.8796
 LOG(SD/MEAN)         -1.3750    -1.2361     -1.1106     -1.0189     -0.9402     -0.8451
```

```
kρ=50.0 n=100
 DISTANCE              0.0        5.0000     10.0000     15.0000     20.0000     25.0000
 TRUE DENSITY         50.0000    41.3370     23.3924      9.1004      2.4514      0.4618
CONSTANT-AREA A/2π=1%
 MAXIMUM              50.0000    43.0000     31.0000     18.0000      9.0000      3.0000
 MINIMUM              29.0000    22.0000     13.0000      4.0000      1.0000      0.0
 MEAN                 38.7000    33.2000     20.6000     10.3000      3.3000      0.6000
 STD DEV               5.2000     5.0000      4.2000      3.2000      1.6000      0.7000
 MEAN/TRUE             0.7740     0.8032      0.8806      1.1318      1.3462      1.2993
 SD/MEAN               0.1344     0.1506      0.2039      0.3107      0.4848      1.1667
 LOG(MEAN/TRUE)       -0.1113    -0.0952     -0.0552      0.0538      0.1291      0.1137
 LOG(SD/MEAN)         -0.8717    -0.8222     -0.6906     -0.5077     -0.3144      0.0669
CONSTANT-AREA KAMB
 MAXIMUM              12.1000    12.1000     11.7000     10.7000      8.4000      5.4000
 MINIMUM              11.5000    11.3000     10.2000      8.7000      5.9000      3.1000
 MEAN                 11.9000    11.8000     11.1000      9.6000      7.1000      4.3000
 STD DEV               0.2000     0.2000      0.4000      0.5000      0.6000      0.5000
 MEAN/TRUE             0.2380     0.2855      0.4745      1.0549      2.8963      9.3114
 SD/MEAN               0.0168     0.0169      0.0360      0.0521      0.0845      0.1163
 LOG(MEAN/TRUE)       -0.6234    -0.5445     -0.3238      0.0232      0.4618      0.9690
 LOG(SD/MEAN)         -1.7745    -1.7709     -1.4433     -1.2833     -1.0731     -0.9345
CONSTANT-AREA A/2π=2%
 MAXIMUM              36.5000    34.0000     23.0000     16.5000      8.0000      3.5000
 MINIMUM              25.0000    22.5000     15.0000      7.5000      1.5000      0.0
 MEAN                 31.3000    28.1000     19.6000     10.8000      4.6000      1.2000
 STD DEV               2.6000     2.8000      2.0000      2.0000      1.4000      0.8000
 MEAN/TRUE             0.6260     0.6798      0.8379      1.1868      1.8765      2.5985
 SD/MEAN               0.0831     0.0996      0.1020      0.1852      0.3043      0.6667
 LOG(MEAN/TRUE)       -0.2034    -0.1676     -0.0768      0.0744      0.2733      0.4147
 LOG(SD/MEAN)         -1.0806    -1.0015     -0.9912     -0.7324     -0.5166     -0.1761
CONSTANT-AREA A/2π=5%
 MAXIMUM              19.6000    19.0000     16.8000     13.0000      8.0000      4.6000
 MINIMUM              16.8000    15.8000     13.6000      9.0000      4.6000      1.2000
 MEAN                 18.3000    17.5000     15.1000     11.0000      6.4000      3.0000
 STD DEV               0.6000     0.8000      0.8000      0.9000      0.8000      0.8000
 MEAN/TRUE             0.3660     0.4233      0.6455      1.2087      2.6108      6.4963
 SD/MEAN               0.0328     0.0457      0.0530      0.0818      0.1250      0.2667
 LOG(MEAN/TRUE)       -0.4365    -0.3733     -0.1901      0.0823      0.4168      0.8127
 LOG(SD/MEAN)         -1.4843    -1.3399     -1.2759     -1.0871     -0.9031     -0.5740
CONSTANT-AREA A/2π=10%
 MAXIMUM              10.0000    10.0000     10.0000      9.3000      7.8000      5.6000
 MINIMUM               9.7000     9.6000      8.9000      7.6000      6.1000      3.9000
 MEAN                  9.9000     9.9000      9.6000      8.7000      7.0000      4.7000
 STD DEV               0.1000     0.1000      0.2000      0.4000      0.5000      0.4000
 MEAN/TRUE             0.1980     0.2395      0.4104      0.9560      2.8555     10.1776
 SD/MEAN               0.0101     0.0101      0.0208      0.0460      0.0714      0.0851
 LOG(MEAN/TRUE)       -0.7033    -0.6207     -0.3868     -0.0195      0.4557      1.0076
 LOG(SD/MEAN)         -1.9956    -1.9956     -1.6812     -1.3375     -1.1461     -1.0700
```

```
kρ=50.0 n=100
 DISTANCE                   0.0        5.0000     10.0000     15.0000     20.0000     25.0000
 TRUE DENSITY              50.0000    41.3370     23.3924      9.1004      2.4514      0.4618
CONST. VAR. 1/3
 MAXIMUM                   80.3000    73.0000     41.1000     23.7000      9.2000      4.1000
 MINIMUM                   26.3000    20.1000     13.4000      5.9000      2.9000      1.8000
 MEAN                      49.5000    37.7000     24.8000     11.2000      5.0000      2.6000
 STD DEV                   14.1000    11.7000      8.4000      3.8000      1.2000      0.5000
 MEAN/TRUE                  0.9900     0.9120      1.0602      1.2307      2.0397      5.6301
 SD/MEAN                    0.2848     0.3103      0.3387      0.3393      0.2400      0.1923
 LOG(MEAN/TRUE)            -0.0044    -0.0400      0.0254      0.0902      0.3096      0.7505
 LOG(SD/MEAN)              -0.5454    -0.5082     -0.4702     -0.4694     -0.6198     -0.7160
CONST. VAR. 1/5
 MAXIMUM                   63.1000    49.3000     42.0000     17.1000      8.4000      4.7000
 MINIMUM                   30.9000    21.6000     14.5000      7.8000      4.3000      2.9000
 MEAN                      44.9000    35.0000     22.0000     11.2000      6.0000      3.6000
 STD DEV                    9.4000     6.5000      5.0000      2.0000      0.9000      0.4000
 MEAN/TRUE                  0.8980     0.8467      0.9405      1.2307      2.4476      7.7956
 SD/MEAN                    0.2094     0.1857      0.2273      0.1786      0.1500      0.1111
 LOG(MEAN/TRUE)            -0.0467    -0.0723     -0.0267      0.0902      0.3887      0.8918
 LOG(SD/MEAN)              -0.6791    -0.7312     -0.6435     -0.7482     -0.8239     -0.9542
CONSTANT-PROPORTION  Vo=1/10
 MAXIMUM                   44.7000    42.2000     23.3000     13.9000      9.1000      6.0000
 MINIMUM                   24.7000    21.8000     13.7000      8.6000      5.8000      4.1000
 MEAN                      34.8000    29.1000     18.5000     11.2000      7.2000      4.9000
 STD DEV                    5.3000     4.2000      2.1000      1.1000      0.6000      0.4000
 MEAN/TRUE                  0.6960     0.7040      0.7909      1.2307      2.9371     10.6107
 SD/MEAN                    0.1523     0.1443      0.1135      0.0982      0.0833      0.0816
 LOG(MEAN/TRUE)            -0.1574    -0.1524     -0.1019      0.0902      0.4679      1.0257
 LOG(SD/MEAN)              -0.8173    -0.8406     -0.9450     -1.0078     -1.0792     -1.0881
CONSTANT-PROPORTION  Vo=1/20
 MAXIMUM                   32.4000    27.3000     16.8000     11.3000      7.6000      5.7000
 MINIMUM                   17.0000    14.9000     10.6000      7.5000      5.5000      4.2000
 MEAN                      24.2000    19.8000     13.7000      9.4000      6.7000      5.0000
 STD DEV                    3.6000     2.6000      1.5000      0.9000      0.5000      0.3000
 MEAN/TRUE                  0.4840     0.4790      0.5857      1.0329      2.7331     10.8272
 SD/MEAN                    0.1488     0.1313      0.1095      0.0957      0.0746      0.0600
 LOG(MEAN/TRUE)            -0.3152    -0.3197     -0.2324      0.0141      0.4367      1.0345
 LOG(SD/MEAN)              -0.8275    -0.8817     -0.9606     -1.0189     -1.1271     -1.2218
```

```
kp =50.0 n=100
 DISTANCE                 0.0       5.0000     10.0000     15.0000     20.0000     25.0000
 TRUE DENSITY         50.0000      41.3370     23.3924      9.1004      2.4514      0.4618
FISHER-WEIGHTED Kw=242
 MAXIMUM              51.9000      44.3000     34.6000     20.0000     10.7000      2.3000
 MINIMUM              28.2000      20.8000     11.3000      2.8000      0.9000      0.0
 MEAN                 40.7000      33.2000     21.6000     10.1000      2.9000      0.5000
 STD DEV               6.1000       5.8000      5.3000      3.6000      1.7000      0.6000
 MEAN/TRUE             0.8140       0.8032      0.9234      1.1098      1.1830      1.0827
 SD/MEAN               0.1499       0.1747      0.2454      0.3564      0.5862      1.2000
 LOG(MEAN/TRUE)       -0.0894      -0.0952     -0.0346      0.0453      0.0730      0.0345
 LOG(SD/MEAN)         -0.8243      -0.7577     -0.6102     -0.4480     -0.2319      0.0792
FISHER-WEIGHTED Kw=100
 MAXIMUM              38.5000      33.3000     27.0000     15.3000      8.6000      3.4000
 MINIMUM              27.3000      22.0000     14.1000      5.7000      2.2000      0.4000
 MEAN                 32.9000      28.6000     19.6000     10.5000      4.2000      1.3000
 STD DEV               3.1000       2.9000      2.6000      2.1000      1.2000      0.6000
 MEAN/TRUE             0.6580       0.6919      0.8379      1.1538      1.7133      2.8151
 SD/MEAN               0.0942       0.1014      0.1327      0.2000      0.2857      0.4615
 LOG(MEAN/TRUE)       -0.1818      -0.1600     -0.0768      0.0621      0.2338      0.4495
 LOG(SD/MEAN)         -1.0258      -0.9940     -0.8773     -0.6990     -0.5441     -0.3358
FISHER-WEIGHTED Kw=50
 MAXIMUM              28.0000      25.4000     20.3000     13.5000      7.6000      3.9000
 MINIMUM              22.1000      19.1000     13.7000      7.6000      3.5000      1.4000
 MEAN                 24.8000      22.4000     16.8000     10.5000      5.4000      2.3000
 STD DEV               1.5000       1.5000      1.4000      1.2000      0.9000      0.5000
 MEAN/TRUE             0.4960       0.5419      0.7182      1.1538      2.2028      4.9805
 SD/MEAN               0.0605       0.0670      0.0833      0.1143      0.1667      0.2174
 LOG(MEAN/TRUE)       -0.3045      -0.2661     -0.1438      0.0621      0.3430      0.6973
 LOG(SD/MEAN)         -1.2184      -1.1742     -1.0792     -0.9420     -0.7782     -0.6628
FISHER-WEIGHTED Kw=25
 MAXIMUM              18.0000      17.0000     14.2000     10.8000      7.1000      4.3000
 MINIMUM              15.4000      14.2000     11.3000      7.9000      4.8000      2.6000
 MEAN                 16.6000      15.5000     12.8000      9.4000      6.0000      3.4000
 STD DEV               0.6000       0.6000      0.7000      0.6000      0.5000      0.4000
 MEAN/TRUE             0.3320       0.3750      0.5472      1.0329      2.4476      7.3625
 SD/MEAN               0.0361       0.0387      0.0547      0.0638      0.0833      0.1176
 LOG(MEAN/TRUE)       -0.4789      -0.4260     -0.2619      0.0141      0.3887      0.8670
 LOG(SD/MEAN)         -1.4420      -1.4122     -1.2621     -1.1950     -1.0792     -0.9294
FISHER-WEIGHTED Kw=500
 MAXIMUM              62.3000      55.7000     39.6000     24.4000     12.3000      1.9000
 MINIMUM              24.8000      14.1000      7.1000      1.2000      0.1000      0.0
 MEAN                 44.5000      34.4000     22.4000      9.7000      2.1000      0.2000
 STD DEV               9.3000       9.0000      8.2000      5.1000      2.4000      0.4000
 MEAN/TRUE             0.8900       0.8322      0.9576      1.0659      0.8567      0.4331
 SD/MEAN               0.2090       0.2616      0.3661      0.5258      1.1429      2.0000
 LOG(MEAN/TRUE)       -0.0506      -0.0798     -0.0188      0.0277     -0.0672     -0.3634
 LOG(SD/MEAN)         -0.6799      -0.5823     -0.4364     -0.2792      0.0580      0.3010
```

```
kp=50.0 n=200
 DISTANCE              0.0        5.0000    10.0000    15.0000    20.0000    25.0000
 TRUE DENSITY         50.0000    41.3370    23.3924     9.1004     2.4514     0.4618
CONSTANT-AREA A/2π=2%
 MAXIMUM              34.2500    31.2500    24.0000    14.2500     6.5000     3.0000
 MINIMUM              27.7500    23.5000    17.5000     7.5000     2.5000     0.2500
 MEAN                 31.6200    28.2850    20.3050    11.1950     4.6350     1.4650
 STD DEV               1.5545     1.5860     1.6693     1.5545     0.9626     0.5670
 MEAN/TRUE             0.6324     0.6843     0.8680     1.2302     1.8908     3.1724
 SD/MEAN               0.0492     0.0561     0.0822     0.1389     0.2077     0.3870
 LOG(MEAN/TRUE)       -0.1990    -0.1648    -0.0615     0.0900     0.2766     0.5014
 LOG(SD/MEAN)         -1.3084    -1.2513    -1.0851    -0.8574    -0.6826    -0.4123
CONSTANT-AREA A/2π=5%
 MAXIMUM              19.4000    18.7000    16.7000    12.7000     8.4000     4.1000
 MINIMUM              17.6000    16.3000    13.5000     9.8000     5.5000     1.9000
 MEAN                 18.3519    17.5519    15.1499    11.1519     6.6540     3.0840
 STD DEV               0.3894     0.4653     0.5700     0.6721     0.6537     0.4752
 MEAN/TRUE             0.3670     0.4246     0.6476     1.2254     2.7144     6.6782
 SD/MEAN               0.0212     0.0265     0.0376     0.0603     0.0982     0.1541
 LOG(MEAN/TRUE)       -0.4353    -0.3720    -0.1887     0.0883     0.4337     0.8247
 LOG(SD/MEAN)         -1.6733    -1.5766    -1.4245    -1.2199    -1.0077    -0.8122
CONSTANT-AREA A/2π=1%
 MAXIMUM              46.0000    39.0000    29.5000    15.5000     6.5000     2.5000
 MINIMUM              32.5000    27.0000    15.5000     7.0000     1.0000     0.0
 MEAN                 39.2000    33.7000    21.7000    10.4000     3.4000     0.8000
 STD DEV               2.9000     2.8000     2.9000     1.9000     1.2000     0.6000
 MEAN/TRUE             0.7840     0.8153     0.9277     1.1428     1.3870     1.7324
 SD/MEAN               0.0740     0.0831     0.1336     0.1827     0.3529     0.7500
 LOG(MEAN/TRUE)       -0.1057    -0.0887    -0.0326     0.0580     0.1421     0.2386
 LOG(SD/MEAN)         -1.1309    -1.0805    -0.8741    -0.7383    -0.4523    -0.1249
CONSTANT-AREA KAMB
 MAXIMUM              22.2000    21.2000    17.5000    13.0000     8.3000     3.8000
 MINIMUM              19.0000    18.3000    14.6000    10.1000     4.8000     1.5000
 MEAN                 20.5000    19.5000    16.1000    11.2000     6.2000     2.6000
 STD DEV               0.5000     0.6000     0.6000     0.6000     0.7000     0.5000
 MEAN/TRUE             0.4100     0.4717     0.6883     1.2307     2.5292     5.6301
 SD/MEAN               0.0244     0.0308     0.0373     0.0536     0.1129     0.1923
 LOG(MEAN/TRUE)       -0.3872    -0.3263    -0.1622     0.0902     0.4030     0.7505
 LOG(SD/MEAN)         -1.6128    -1.5119    -1.4287    -1.2711    -0.9473    -0.7160
```

```
k0=50.0 n=200
 DISTANCE                 0.0       5.0000     10.0000     15.0000     20.0000     25.0000
 TRUE DENSITY          50.0000     41.3370     23.3924      9.1004      2.4514      0.4618
CONSTANT-PROPORTION  V0 =1/7
 MAXIMUM               60.4423     50.2000     28.8443     14.1514      7.1902      4.2283
 MINIMUM               28.8443     26.2025     16.2448      8.9739      4.6916      3.0112
 MEAN                  45.2508     38.1012     22.3301     11.4716      6.0186      3.6560
 STD DEV                7.3625      5.3662      2.9690      1.4974      0.6160      0.2737
 MEAN/TRUE              0.9050      0.9217      0.9546      1.2606      2.4552      7.9168
 SD/MEAN                0.1627      0.1408      0.1330      0.1305      0.1023      0.0749
 LOG(MEAN/TRUE)        -0.0433     -0.0354     -0.0202      0.1006      0.3901      0.8986
 LQG(SD/MEAN)          -0.7886     -0.8513     -0.8763     -0.8843     -0.9899     -1.1257
CONSTANT-PROPORTION  V0 =1/10
 MAXIMUM               49.6391     46.9559     25.6711     13.8510      8.4792      5.0126
 MINIMUM               28.8345     26.6584     16.9318      9.5513      5.7207      3.8424
 MEAN                  41.3600     34.3982     20.9156     11.6217      6.8696      4.40[illegible]
 STD DEV                4.9129      3.9306      2.0685      0.9890      0.5289      0.2838
 MEAN/TRUE              0.8272      0.8321      0.8941      1.2771      2.8023      9.5390
 SD/MEAN                0.1188      0.1143      0.0989      0.0851      0.0770      0.0644
 LOG(MEAN/TRUE)        -0.0824     -0.0798     -0.0486      0.1062      0.4475      0.9795
 LOG(SD/MEAN)          -0.9252     -0.9421     -1.0048     -1.0701     -1.1136     -1.1909
CONSTANT-PROPORTION  V0 =1/20
 MAXIMUM               37.0113     28.0922     19.8098     12.4813      8.1546      5.7239
 MINIMUM               23.9023     20.6902     13.6893      8.8685      6.2389      4.5409
 MEAN                  29.3847     24.9700     16.4549     10.6184      7.1991      5.1525
 STD DEV                2.4325      2.0034      1.2791      0.7216      0.4058      0.2431
 MEAN/TRUE              0.5877      0.6041      0.7034      1.1668      2.9367     11.1574
 SD/MEAN                0.0828      0.0802      0.0777      0.0680      0.0564      0.0472
 LOG(MEAN/TRUE)        -0.2308     -0.2189     -0.1528      0.0670      0.4679      1.0476
 LOG(SD/MEAN)          -1.0821     -1.0957     -1.1094     -1.1678     -1.2490     -1.3262
CONSTANT-PROPORTION  V0 =1/3
 MAXIMUM               98.4000    106.7000     47.4000     17.0000      6.7000      2.7000
 MINIMUM               28.7000     20.5000     11.8000      6.0000      2.4000      1.2000
 MEAN                  51.8000     43.2000     25.2000     10.4000      4.0000      1.8000
 STD DEV               15.4000     14.2000      7.7000      2.7000      0.8000      0.3000
 MEAN/TRUE              1.0360      1.0451      1.0773      1.1428      1.6317      3.8978
 SD/MEAN                0.2973      0.3287      0.3056      0.2596      0.2000      0.1667
 LOG(MEAN/TRUE)         0.0154      0.0191      0.0323      0.0580      0.2126      0.5908
 LOG(SD/MEAN)          -0.5268     -0.4832     -0.5149     -0.5857     -0.6990     -0.7782
CONSTANT-PROPORTION  V0 =1/5
 MAXIMUM               69.7000     55.4000     39.4000     15.4000      6.8000      3.6000
 MINIMUM               34.1000     26.4000     15.9000      7.5000      3.8000      2.2000
 MEAN                  48.3000     40.7000     23.9000     10.7000      5.1000      2.8000
 STD DEV                7.7000      7.9000      5.0000      1.7000      0.6000      0.3000
 MEAN/TRUE              0.9660      0.9846      1.0217      1.1758      2.0804      6.0632
 SD/MEAN                0.1594      0.1941      0.2092      0.1589      0.1176      0.1071
 LOG(MEAN/TRUE)        -0.0150     -0.0067      0.0093      0.0703      0.3182      0.7827
 LOG(SD/MEAN)          -0.7975     -0.7120     -0.6794     -0.7989     -0.9294     -0.9700
```

```
kρ=50.0 n=200
 DISTANCE                0.0        5.0000     10.0000     15.0000     20.0000     25.0000
 TRUE DENSITY           50.0000    41.3370     23.3924      9.1004      2.4514      0.4618
FISHER-WEIGHTED Kw=242
 MAXIMUM                51.7000    44.5000     33.4000     14.4000      6.4000      3.0000
 MINIMUM                31.9000    25.8000     14.7000      5.9000      0.7000      0.0
 MEAN                   41.4000    35.4000     22.3000      9.9000      3.1000      0.7000
 STD DEV                 3.7000     4.0000      3.4000      2.2000      1.2000      0.6000
 MEAN/TRUE               0.8280     0.8564      0.9533      1.0879      1.2646      1.5158
 SD/MEAN                 0.0894     0.1130      0.1525      0.2222      0.3871      0.8571
 LOG(MEAN/TRUE)         -0.0820    -0.0673     -0.0208      0.0366      0.1019      0.1806
 LOG(SD/MEAN)           -1.0488    -0.9469     -0.8168     -0.6532     -0.4122     -0.0669
FISHER-WEIGHTED Kw=100
 MAXIMUM                37.9000    33.1000     25.8000     14.3000      5.9000      2.5000
 MINIMUM                28.5000    24.7000     16.3000      8.2000      2.5000      0.4000
 MEAN                   33.4000    29.4000     20.1000     10.6000      4.3000      1.3000
 STD DEV                 1.8000     1.7000      1.7000      1.3000      0.8000      0.4000
 MEAN/TRUE               0.6680     0.7112      0.8593      1.1648      1.7541      2.8151
 SD/MEAN                 0.0539     0.0578      0.0846      0.1226      0.1860      0.3077
 LOG(MEAN/TRUE)         -0.1752    -0.1480     -0.0659      0.0662      0.2441      0.4495
 LOG(SD/MEAN)           -1.2685    -1.2379     -1.0727     -0.9114     -0.7304     -0.5119
FISHER-WEIGHTED Kw=50
 MAXIMUM                27.4000    24.9000     19.6000     13.0000      6.9000      3.0000
 MINIMUM                22.7000    20.6000     15.4000      9.2000      4.2000      1.5000
 MEAN                   25.0000    22.7000     17.1000     10.6000      5.4000      2.3000
 STD DEV                 0.9000     0.8000      0.9000      0.8000      0.6000      0.4000
 MEAN/TRUE               0.5000     0.5491      0.7310      1.1648      2.2028      4.9805
 SD/MEAN                 0.0360     0.0352      0.0526      0.0755      0.1111      0.1739
 LOG(MEAN/TRUE)         -0.3010    -0.2603     -0.1361      0.0662      0.3430      0.6973
 LOG(SD/MEAN)           -1.4437    -1.4529     -1.2788     -1.1222     -0.9542     -0.7597
FISHER-WEIGHTED Kw=25
 MAXIMUM                17.7000    16.7000     13.8000     10.5000      7.0000      4.1000
 MINIMUM                15.7000    14.8000     12.2000      8.6000      5.3000      2.8000
 MEAN                   16.7000    15.6000     12.9000      9.4000      6.0000      3.4000
 STD DEV                 0.3000     0.3000      0.3000      0.4000      0.3000      0.2000
 MEAN/TRUE               0.3340     0.3774      0.5515      1.0329      2.4476      7.3625
 SD/MEAN                 0.0180     0.0192      0.0233      0.0426      0.0500      0.0588
 LOG(MEAN/TRUE)         -0.4763    -0.4232     -0.2585      0.0141      0.3887      0.8670
 LOG(SD/MEAN)           -1.7456    -1.7160     -1.6335     -1.3711     -1.3010     -1.2304
```

```
kρ=100.0 n=100
 DISTANCE              0.0       5.0000    10.0000    15.0000    20.0000    25.0000
 TRUE DENSITY        100.0000   68.3500    21.8882     3.3127     0.2404     0.0085
CONSTANT-AREA A/2π=2%
 MAXIMUM              46.5000   41.5000    31.0000    15.5000     4.5000     1.0000
 MINIMUM              39.0000   32.5000    20.0000     5.5000     0.5000     0.0
 MEAN                 43.3000   37.8700    24.6400    10.0600     2.2800     0.2500
 STD DEV               1.6413    1.7317     2.6689     2.0016     0.9322     0.3388
 MEAN/TRUE             0.4330    0.5541     1.1257     3.0368     9.4842    29.4118
 SD/MEAN               0.0379    0.0457     0.1083     0.1990     0.4089     1.3552
 LOG(MEAN/TRUE)       -0.3635   -0.2564     0.0514     0.4824     0.9770     1.4685
 LOG(SD/MEAN)         -1.4213   -1.3398    -0.9653    -0.7012    -0.3884     0.1320
CONSTANT-AREA A/2π=5%
 MAXIMUM              20.0000   20.0000    19.0000    15.2000     9.0000     3.2000
 MINIMUM              19.4000   19.0000    16.4000    10.6000     5.2000     1.0000
 MEAN                 19.8679   19.5279    17.8199    12.9999     6.5240     1.8440
 STD DEV               0.1585    0.2708     0.6087     1.0083     0.8192     0.5589
 MEAN/TRUE             0.1989    0.2857     0.8141     3.9243    27.1381   216.9412
 SD/MEAN               0.0080    0.0139     0.0342     0.0776     0.1256     0.3031
 LOG(MEAN/TRUE)       -0.7014   -0.5441    -0.0893     0.5938     1.4336     2.3363
 LOG(SD/MEAN)         -2.0986   -1.8580    -1.4665    -1.1103    -0.9011    -0.5184
CONSTANT-AREA A/2π=1%
 MAXIMUM              73.0000   62.0000    34.0000    16.0000     5.0000     1.0000
 MINIMUM              54.0000   41.0000    14.0000     2.0000     0.0        0.0
 MEAN                 63.6000   50.9000    24.9000     7.8000     1.3000     0.1000
 STD DEV               4.8000    4.2000     4.8000     2.9000     1.1000     0.3000
 MEAN/TRUE             0.6360    0.7447     1.1376     2.3546     5.4077    11.7647
 SD/MEAN               0.0755    0.0825     0.1928     0.3718     0.8462     3.0000
 LOG(MEAN/TRUE)       -0.1965   -0.1280     0.0560     0.3719     0.7330     1.0706
 LOG(SD/MEAN)         -1.1222   -1.0835    -0.7150    -0.4297    -0.0726     0.4771
CONSTANT-AREA KAMB
 MAXIMUM              12.1000   12.1000    12.1000    11.8000     9.6000     5.3000
 MINIMUM              12.0000   12.0000    11.5000    10.4000     7.2000     2.4000
 MEAN                 12.1000   12.0000    11.9000    11.0000     8.3000     4.2000
 STD DEV               0.0       0.0        0.1000     0.3000     0.4000     0.6000
 MEAN/TRUE             0.1210    0.1756     0.5437     3.3206    34.5258   494.1177
 SD/MEAN               0.0       0.0        0.0084     0.0273     0.0482     0.1429
 LOG(MEAN/TRUE)       -0.9172   -0.7556    -0.2647     0.5212     1.5381     2.6938
 LOG(SD/MEAN)        -10.0000  -10.0000    -2.0755    -1.5643    -1.3170    -0.8451
```

```
kp=100.0 n=100
 DISTANCE               0.0       5.0000     10.0000     15.0000     20.0000     25.0000
 TRUE DENSITY         100.0000   68.3500     21.8882      3.3127      0.2404      0.0085
CONSTANT-PROPORTION V0 =1/7
 MAXIMUM              155.9984   74.8437     35.9642     15.6068      8.0762      4.8580
 MINIMUM               56.3471   44.2727     18.5056      9.6848      5.9694      3.8170
 MEAN                  80.9046   57.4448     25.8164     12.2208      6.8232      4.3930
 STD DEV               16.2389    8.0418      3.5592      1.2343      0.5210      0.2865
 MEAN/TRUE              0.8090    0.8405      1.1795      3.6891     28.3827    516.8235
 SD/MEAN                0.2007    0.1400      0.1379      0.1010      0.0764      0.0652
 LOG(MEAN/TRUE)        -0.0920   -0.0755      0.0717      0.5669      1.4531      2.7133
 LOG(SD/MEAN)          -0.6974   -0.8539     -0.8605     -0.9957     -1.1171     -1.1856
CONSTANT-PROPORTION V0 =1/10
 MAXIMUM               96.7491   69.7631     31.5684     16.0049      9.1200      6.1905
 MINIMUM               53.4326   36.2028     19.1892     10.6245      6.9187      4.6792
 MEAN                  67.9624   49.6058     24.7033     13.1496      7.9583      5.3308
 STD DEV                8.3673    6.1682      2.8381      1.2405      0.5558      0.3524
 MEAN/TRUE              0.6796    0.7258      1.1286      3.9695     33.1044    627.1528
 SD/MEAN                0.1231    0.1243      0.1149      0.0943      0.0698      0.0661
 LOG(MEAN/TRUE)        -0.1677   -0.1392      0.0525      0.5987      1.5199      2.7974
 LOG(SD/MEAN)          -0.9097   -0.9054     -0.9397     -1.0253     -1.1559     -1.1798
CONSTANT-PROPORTION V0 =1/20
 MAXIMUM               56.6080   38.8975     22.2003     13.7628      9.4848      6.6199
 MINIMUM               37.5246   24.9456     16.5497     10.5667      7.2617      5.3964
 MEAN                  47.2800   34.0660     19.8419     12.2503      8.2005      5.8615
 STD DEV                5.2999    3.3695      1.3916      0.7457      0.4558      0.2940
 MEAN/TRUE              0.4728    0.4984      0.9065      3.6980     34.1119    689.5881
 SD/MEAN                0.1121    0.0989      0.0701      0.0609      0.0556      0.0502
 LOG(MEAN/TRUE)        -0.3253   -0.3024     -0.0426      0.5680      1.5329      2.8386
 LOG(SD/MEAN)          -0.9504   -1.0048     -1.1541     -1.2156     -1.2551     -1.2997
CONSTANT-PROPORTION V0 =1/3
 MAXIMUM              180.5000  131.3000     65.6000     16.0000      5.4000      2.8000
 MINIMUM               49.2000   34.1000     12.0000      5.9000      2.6000      1.4000
 MEAN                  97.6000   67.5000     28.0000      8.8000      3.8000      2.1000
 STD DEV               33.4000   21.5000      9.6000      1.9000      0.6000      0.3000
 MEAN/TRUE              0.9760    0.9876      1.2792      2.6564     15.8070    247.0589
 SD/MEAN                0.3422    0.3185      0.3429      0.2159      0.1579      0.1429
 LOG(MEAN/TRUE)        -0.0106   -0.0054      0.1069      0.4243      1.1988      2.3928
 LOG(SD/MEAN)          -0.4657   -0.4969     -0.4649     -0.6657     -0.8016     -0.8451
CONSTANT-PROPORTION V0 =1/5
 MAXIMUM              127.2000  102.6000     38.3000     13.8000      6.7000      4.0000
 MINIMUM               57.8000   36.5000     17.0000      6.7000      4.2000      2.7000
 MEAN                  88.3000   61.0000     25.6000     10.7000      5.4000      3.3000
 STD DEV               16.0000   14.1000      4.9000      1.6000      0.6000      0.3000
 MEAN/TRUE              0.8830    0.8925      1.1696      3.2300     22.4626    388.2354
 SD/MEAN                0.1812    0.2311      0.1914      0.1495      0.1111      0.0909
 LOG(MEAN/TRUE)        -0.0540   -0.0494      0.0680      0.5092      1.3515      2.5891
 LOG(SD/MEAN)          -0.7418   -0.6361     -0.7180     -0.8253     -0.9542     -1.0414
```

```
kρ=100.0 n=100
 DISTANCE              0.0        5.0000    10.0000    15.0000    20.0000    25.0000
 TRUE DENSITY        100.0000    68.3500    21.8882     3.3127     0.2404     0.0085
FISHER-WEIGHTED Kw=242
 MAXIMUM              84.4000    73.2000    35.3000    14.5000     3.5000     1.3000
 MINIMUM              54.5000    38.4000    16.1000     2.0000     0.0        0.0
 MEAN                 70.2000    53.6000    25.0000     6.9000     1.0000     0.1000
 STD DEV               6.3000     7.4000     4.9000     2.5000     0.8000     0.2000
 MEAN/TRUE             0.7020     0.7842     1.1422     2.0829     4.1597    11.7647
 SD/MEAN               0.0897     0.1381     0.1960     0.3623     0.8000     2.0000
 LOG(MEAN/TRUE)       -0.1537    -0.1056     0.0577     0.3187     0.6191     1.0706
 LOG(SD/MEAN)         -1.0470    -0.8599    -0.7077    -0.4409    -0.0969     0.3010
FISHER-WEIGHTED Kw=100
 MAXIMUM              55.8000    48.0000    28.4000    14.1000     5.0000     1.2000
 MINIMUM              43.4000    34.6000    17.5000     5.9000     1.1000     0.1000
 MEAN                 49.9000    41.4000    23.7000     9.4000     2.5000     0.4000
 STD DEV               2.5000     3.0000     2.6000     1.7000     0.7000     0.2000
 MEAN/TRUE             0.4990     0.6057     1.0828     2.8376    10.3993    47.0588
 SD/MEAN               0.0501     0.0725     0.1097     0.1809     0.2800     0.5000
 LOG(MEAN/TRUE)       -0.3019    -0.2177     0.0345     0.4529     1.0170     1.6726
 LOG(SD/MEAN)         -1.3002    -1.1399    -0.9598    -0.7427    -0.5528    -0.3010
FISHER-WEIGHTED Kw=50
 MAXIMUM              35.7000    32.1000    22.7000    13.4000     6.4000     2.4000
 MINIMUM              30.5000    26.4000    17.0000     8.3000     3.3000     0.9000
 MEAN                 33.3000    29.4000    20.3000    10.9000     4.5000     1.5000
 STD DEV               1.0000     1.2000     1.3000     1.1000     0.7000     0.3000
 MEAN/TRUE             0.3330     0.4301     0.9274     3.2904    18.7188   176.4706
 SD/MEAN               0.0300     0.0408     0.0640     0.1009     0.1556     0.2000
 LOG(MEAN/TRUE)       -0.4776    -0.3664    -0.0327     0.5172     1.2723     2.2467
 LOG(SD/MEAN)         -1.5224    -1.3892    -1.1936    -0.9960    -0.8081    -0.6990
FISHER-WEIGHTED Kw=25
 MAXIMUM              20.8000    19.5000    15.9000    11.3000     7.1000     3.8000
 MINIMUM              19.0000    17.4000    13.5000     8.8000     5.0000     2.5000
 MEAN                 20.0000    18.5000    14.8000    10.2000     6.0000     3.1000
 STD DEV               0.3000     0.4000     0.5000     0.5000     0.4000     0.3000
 MEAN/TRUE             0.2000     0.2707     0.6762     3.0791    24.9584   364.7058
 SD/MEAN               0.0150     0.0216     0.0338     0.0490     0.0667     0.0968
 LOG(MEAN/TRUE)       -0.6990    -0.5676    -0.1699     0.4884     1.3972     2.5619
 LOG(SD/MEAN)         -1.8239    -1.6651    -1.4713    -1.3096    -1.1761    -1.0142
```